Computers in Biology and Medicine 146 (2022) 105645

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Uncertainty-aware selecting for an ensemble of deep food recognition models

Eduardo Aguilar [a,*], Bhalaji Nagarajan [b], Petia Radeva [b]

[a] Department of Computing and Systems Engineering, Catholic University of the North, Avenida Angamos 0610, Antofagasta, 1270709, Antofagasta, Chile
[b] Department of Mathematics and Computer Science, University of Barcelona, Gran Via de les Corts Catalanes 585, Barcelona, 08007, Barcelona, Spain

ARTICLE INFO

Keywords:
Food recognition deep learning ensemble learning model selection convolutional neural networks uncertainty estimation

ABSTRACT

Deep learning is a machine learning technique that has revolutionized the research community due to its impressive results on various real-life problems. Recently, ensembles of Convolutional Neural Networks (CNN) have proven to achieve high robustness and accuracy in numerous computer vision challenges. As expected, the more models we add to the ensemble, the better performance we can obtain, but, in contrast, more computer resources are needed. Hence, the importance of deciding how many models to use and which models to select from a pool of trained models is huge. From the latter, a common strategy in deep learning is to select the models randomly or according to the results on the validation set. However, in this way models are chosen based on individual performance ignoring how they are expected to work together. Alternatively, to ensure a better complement between models, an exhaustive search can be used by evaluating the performance of several ensemble models based on different numbers and combinations of trained models. Nevertheless, this may result in being high computationally expensive. Considering that epistemic uncertainty analysis has recently been successfully employed to understand model learning, we aim to analyze whether an uncertainty-aware epistemic method can help us decide which groups of CNN models may work best. The method was validated on several food datasets and with different CNN architectures. In most cases, our proposal outperforms the results by a statistically significant range with respect to the baseline techniques and is much less computationally expensive compared to the brute-force search.

1. Introduction

Healthy-eating has become one of the buzzwords in the modern society. Nutritional patterns [1] are a key factor in maintaining the well-being of individuals and of late, more and more health platforms [2] have become very popular and have been instrumental in maintaining the dietary intake of people. With the fast-paced development in the fields of computer vision and deep learning, image-based food journals have become the latest trend [3]. Automated food recognition is an integral component to these food journals [4]. However, food recognition is an onerous computer vision task, owing to the nature and complexity of food images [5]. The main factors that contribute to the complexity of food image analysis are the shape, volume, texture, color and composition of food classes that can be represented in numerous ways [6]. In addition to this, the background, the combination and the layout of food dishes make automated food recognition highly complicated. Also, food images suffer from high intra-class variance and low inter-class variance. Fig. 1 shows some sample images that highlight the complexity of food images often used in food recognition.

Convolutional Neural Networks (CNNs) allow to solve many complex computer vision tasks on par with human performance and surpass machine learning based approaches [7]. However, one of the important reasons for a well-trained deep learning model is the large volume of training data. The datasets should be able to provide sufficient training samples representing the wide variability of the task. Several public food recognition datasets [8–11] are available that can be used to train such models. However, these datasets are often from a constrained environment or represent only a particular cuisine. Food images are very common in social platforms such as Facebook and Instagram due to the increased number of posts at various eating places and also on various food recipe websites. However, a considerable effort is needed in making these data trainable and therefore there are a very few datasets having large number of food images [12,13] available to the research community. It has to be noted that available datasets cover still considerably

* Corresponding author.
E-mail address: eaguilar02@ucn.cl (E. Aguilar).

https://doi.org/10.1016/j.compbiomed.2022.105645
Received 27 January 2022; Received in revised form 4 May 2022; Accepted 14 May 2022
Available online 21 May 2022

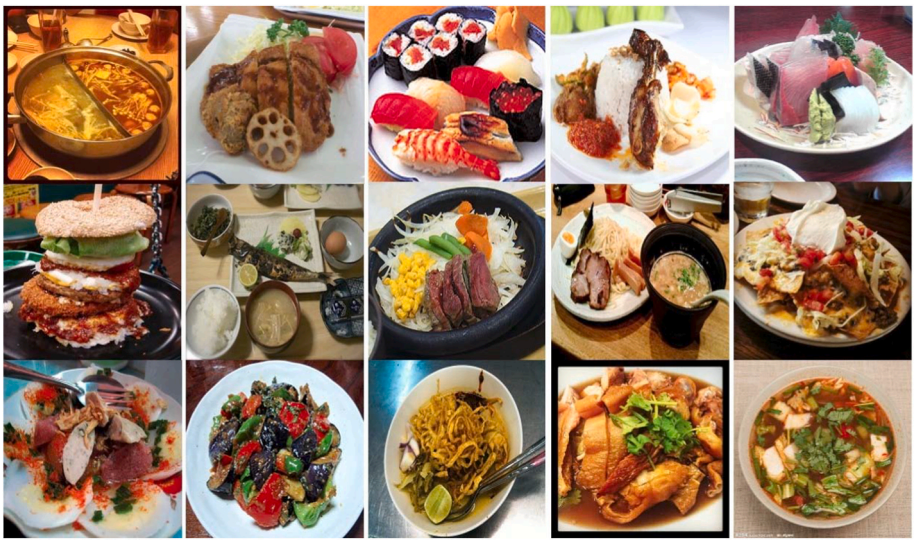

Fig. 1. Sample images showing the complexity of food.

a small percentage of the common food items across the world. A real-world food recognition model would be expected to work on data other than those that are represented by the training data, which sheds importance on the model that is used in the task.

Several popular CNN architectures, such as, AlexNet [14], GoogLeNet [15,16], Inception [17] and Residual Neural Network (ResNet) [18] have been directly adapted for food recognition. However, these direct adaptations provide limited results. With increased computational capabilities and more powerful architectures, ensemble of various models is a common approach [19–21]. Ensembling reduces the overfitting of models and also aids in improving the performance of food recognition [21]. When different models are used on the same task, it can be observed that the patterns of each architecture differ and therefore the results are not always overlapping [22]. This implies that ensembles provide complimentary information, a fact that when combined boosts the overall performance of the models.

In this work, we propose to use epistemic uncertainty [23] in selecting the ensembles in order to achieve higher performance with a previously limited number of base classifiers, subject to computational limitations in real-world solutions. The epistemic uncertainty analysis provides a good clue as to the confidence of the model with respect to its prediction, a factor that has not been fully exploited for the selection of base classifiers. Our contributions are the followings:

Table 1
A summary of deep learning-based work related to visual food analysis.

Reference	Year	Classification Method	Learning Approach	Tasks Addressed	Food Datasets
[16]	2016	CNN	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256
[10]	2016	CNN	Multi-Task	Food and Ingredient Recognition	VIREO172
[20]	2017	Deep network combination (AlexNet, GoogLeNet, ResNet)	Ensemble	Food Recognition	Food-101, Indian Food Database
[21]	2017	Decision Template Scheme (ResNet, Inception)	Ensemble	Food Recognition	Food-11, Food-101
[28]	2018	Comparative of several classifier using feature extracted from ResNet and GoogLeNet	Single-Task	Food Recognition	Food-5K, Food-11, Food-101
[29]	2018	Wide-Slice Residual Network (WISeR)	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256
[30]	2018	CleanNet	Single-Task	Food Recognition	Food-101
[36]	2018	CNN with Spatial Pyramid Pooling	Single-Task	Food Recognition	Food-101, UECFood256
[14]	2019	NutriNet (AlexNet-based CNN)	Single-Task	Food and Beverage Recognition	UNIMIB2016, Private dataset
[34]	2019	Ingredient-Guided Cascaded Multi-Attention Network (IG-CMAN)	Multi-Task	Food and Ingredient Recognition	Food-101, VIREO172, ISIA-Food200
[63]	2019	Regularized Uncertainty-based Multi-Task Learning (RUMTL). It was validated using ResNet50	Multi-Task	Food, Ingredient and Cuisine Style Recognition	MAFood121, VIREO172
[31]	2020	Class Incremental Extreme Learning Machine using features selected from those extracted with InceptionResNetV2	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256, PFID, PAKISTAN FOOD
[62]	2020	Uncertainty-based Hybrid Classification. It was validated using ResNet50	Single-Task	Food Recognition	Food-101, MAFood121
[38]	2020	Joint-learning Distilled Network (JDNet)	Teacher-Student	Food Recognition	Food-101, UECFood256
[19]	2020	Voting-based fine-tuned CNNs (VGG, GoogLeNet, ResNet, Inception)	Ensemble	Food Recognition	Food-101, UECFood100, UECFood256
[35]	2021	Graph Convolutional Network (GCN)	Single-Task	Food Recognition	Food-101, UECFood256
[39]	2021	TL-Ensemble (Soft-Voting in fine-tuned InceptionV3 and DenseNet)	Ensemble	Food Recognition	MAFood121, Food24
[41]	2021	Snapshot Ensembling with MobileNetV3	Ensemble	Food Recognition	Food-101, UECFood100, UECFood256, Malaysian food

- We propose the first Ensemble Selection method to perform the visual food recognition task by mean of CNNs.
- We present a novel measure to perform the Ensemble Selection based on the analysis of the epistemic uncertainty.
- The advantage of the proposed approach was demonstrated in several datasets compared to the baseline approach showing improving of the accuracy meanwhile keeping reasonably fast final food classification.

The remainder of this manuscript is organized as follows: Previous related works are reviewed in the next section. The proposed technique is discussed in Section 3. In Section 4, we introduce the experimental setup and discuss the results obtained in each of the experiments. Finally, conclusions and future research are presented in the last section.

2. Related work

The literature related to the proposed method is briefly discussed covering three fields of research, namely: 1) Food Recognition, 2) Uncertainty Modeling and 3) Ensemble Selection. In addition, the closest visual food analysis works are summarized in Table 1 that indicates the classification method, the learning approach, the task to be solved and the dataset where the method was validated.

2.1. Food recognition

Automated food recognition algorithms have been crucial and central to several food computing tasks, such as in dietary assessment, food perception and food recommendation [4]. Classical machine learning approaches involve segmentation, feature extraction, feature selection or reduction, texture information and classification stages [8,24–26]. This approach was superseded by deep networks after the successful demonstration of their effectiveness in various application areas. Most existing literature on food recognition uses a standard CNN architecture on the food datasets by fine-tuning the model on food data [8,16,27]. Another approach is using the latent space from different layers of CNNs to train different classification algorithms [28]. All these techniques have been very effective on simpler food images, that is, in images acquired in a controlled environment and particularly for those images belonging to a class of foods that show little intra-class variability and little similarity between classes (e.g. edamame).

The recent research in food recognition focused on more advanced deep network architectures. Wide Residual Networks [29] were used to extract the common vertical food characteristics (e.g. vertical arrangement of ingredients in hamburgers) and they were used to improve the classification network. The architecture label cleaning network (CleanNet) [30] was used to handle noisy food data, present when data is automatically collected from the web without manual verification, which is a recurring problem in a complex task such as food recognition. NutriNet [14] was proposed to handle larger image resolutions (512×512) by modifying the standard AlexNet architecture to extract more knowledge from higher-resolution data. Class Incremental Extreme Learning Machines [31] was used in order to handle the open-ended and dynamic nature of the food datasets, which is reflected in an increasing number of food images and food classes that can be generated by creating new food recipes or by different ways of serving the same kind of food. Additional information such as GPS and restaurant information was combined to improve the classification performance [32,33]. Ingredients were also used as supportive information [34] to aid the classifiers in order to better learn food data. The semantic embeddings are found to be correlated with the image representations and this relation has been exploited using Graph Convolution Networks [35].

Another vertical in food recognition research is based on creating models that have less footprint as most of the real-world applications are in mobile platform. An optimized CNN model with spatial pyramid pooling layers [36] was used to reduce the architecture size without compensating on the performance. Network-In-Network [37] architectures are used in real time mobile food recognition apps due to their ease in adapting to the mobile platform and the trade-off they provide. Teacher-student learning is an approach carried out to decrease the full-sized networks. A Joint-Learning Distilled Network (JDNet) [38] used a joint learning framework where both the teacher and the student networks were trained simultaneously.

As it can be seen from the research directions on food recognition, it is evident that attaining higher accurate results is very mandatory. It is due to the direct impact on real-world applications. Ensemble of Deep Networks is a common technique used to improve the performance of algorithms and also to avoid over fitting of networks. FoodNet [20] used the latent space from AlexNet, GoogLeNet and ResNet architectures. Each of the networks was trained separately on food images and then the extracted features were fused together. Tasci in Ref. [19] further enhanced this idea by adding voting combinations for the output of the network architectures which boosted the performance further. Decision Templates [21] were used in fusing the features of ResNet and Inception architectures. TL_Ensemble [39] considered a soft-voting approach (average voting) to integrate the prediction of two different CNN architectures: Inception and Densely Connected Convolutional Networks (DenseNet) [40]. Recently, a method combining the snapshot ensembles approach with MobileNetV3 proved effective for analyzing food images on devices with limited resources [41].

2.2. Uncertainty modeling

There is no doubt that there is *a before* and *after* in image understanding solutions with the arrival of CNNs. The Big Data era together with an increase in computational capacities (e.g. GPU, processor and memory) are enabling factors that have allowed to create deep CNN solutions for various real-world problems. The industry has also been captivated by these solutions, increasingly incorporating them into its business. However, it is not *all rosy in the garden*. Although the Deep Learning algorithms have provided successful results in the experimental condition, when the solution is applied in real-world conditions, in some cases it does not behave as expected. Uncertainty modeling is a research field that tries to understand why models might or might not work. There are two categories of uncertainty [23]: a) aleatory, representing an unexpected behavior of the model produced by noise that could be present in the observation (e.g. noise in the measuring instrument) and b) epistemic, representing an unexpected behavior of the model produced by the lack of sufficient data to achieve a correct fit of the distribution and, therefore, a better generalization. The estimation of the uncertainty can be carried out using a Bayesian neural network (BNN) approach, but in the case of deep learning algorithms, which are characterized by having several parameters and using a large amount of data for their training, it becomes intractable [42–44]. Instead, a stochastic variational inference has been used frequently to capture uncertainty [42,43,45–47]. Of all of them, Monte Carlo (MC)-dropout is the simplest one, but not less efficient method to estimate the uncertainty that is equivalent to using Bernoulli to approximate a variational inference in the BNN [48].

Lately, novel uncertainty-based methods have been proposed to successfully solve a wide range of computer vision tasks. In image classification, uncertainty modeling has been effective in improving classification on unbalanced datasets [49], robustness against adversarial attacks [50] and robustness against noise in face data [51]. For image segmentation, it has been useful for domain adaption [52], weighting the predictions from a multi-view inputs from a 3D data [53], and to create a metric and annotations to solve the night-time semantic segmentation through a curriculum model adaptation from the correspondence of day-night images content [54]. For object detection, it has been applied to model bounding boxes [55,56] and improve the 3D pedestrian location [57]; as to many more (e.g. RGB-D saliency detection [58], facial landmark locations [59], multi-modal learning [60],

image processing [61], etc). Despite the growing interest in uncertainty-based models, we have no evidence that they have been used for food analysis other than ourselves.

In our previous works relating to uncertainty-aware techniques, we have introduced various strategies to improve the performance of food recognition models. Uncertainty-based hierarchical classifiers were used to consider the local classifier performance along with the flat classifier and it showed improved results on several food datasets [62]. Multi-task framework based on homoscedastic uncertainty was used to predict different food-related tasks [63]. Sample-level uncertainty was also used to improve the data augmentation that in turn offered improved performance [64]. In this work, we further investigate the role of uncertainty in selecting the ensemble of several models in order to improve the performance of the food recognition task.

2.3. Ensemble selection

The complex visual appearance of food images results in a real challenge for the computer vision community to analyze them. Recent performance achieved through deep learning algorithms is encouraging, but far from accurate for real-life food recognition problems where thousands of dishes need to be classified. Ensemble learning becomes mandatory to ensure more robust and accurate classification than individual models [65] to address this task. Most of the traditional Ensemble Learning methods can be mainly grouped into three different strategies: bagging [66], boosting [67] and stacking [68]. Bagging methods consist of combining the prediction of a committee of trained models with different subsets of the training data produced by sampling with replacement. Boosting methods involve incorporating one model at a time into the ensemble in order to reduce misclassification of previous models. And Stacking methods work by weighting the predictions given by the committee members through a machine learning algorithm (e.g. logistic regression) trained for that purpose. Particularly for the object recognition task, the common strategy adopted to form an ensemble is through a committee of neural networks, which consists simply of a collection of the same CNN models trained on the same data by varying the weights initialization [69–71]. With this approach, final predictions are calculated by averaging the output of all models.

In some real-world solutions, both accuracy and computational resources are important factors to consider when designing the classification method. Ensemble Learning allows us to improve the accuracy, but also increases the prediction time or the computational resource required (e.g. models running in parallel to perform recognition). Therefore, it is desirable to find a trade-off between the number of committee members with respect to the solution cost. Ensemble Selection (ES) [72,73], also known as ensemble pruning or selective ensemble, is a research field that deals with alleviating the problem of indistinctly considering all base classifiers (members) to build an ensemble.

There are two approaches to find a suitable subset of classifiers [74]: a) Static Ensemble Selection (SES), where all unseen data are classified with the ensemble selected in the training phase; and b) Dynamic Ensemble Selection (DES), which attempts to find the most competent classifiers for each unseen data. Regarding SES, one of the earliest methods in the literature corresponds to a forward step-wise selection approach to perform the ensemble selection, which consists of a greedy procedure iteratively adding models. Each next model tries to optimize the performance metric (e.g. accuracy, root-mean-squared-error, among other) on the validation set, in order to fill an initially empty ensemble [73]. From there, different selection criteria have been proposed to greedy select the ensemble [75–78]. A diversity measure dubbed Uncertainty Weighted Accuracy (UWA) in Refs. [75,76] takes into account the contribution of four events that represent the pairwise combinations given from the classification results of the candidate model and the current ensemble. In addition, they weight the events according to the uncertainty on the ensemble's decision, calculated as the proportion of models within the ensemble that correctly or incorrectly classifies a particular example. Rather than using a single criterion such as accuracy or diversity, recent approaches [77,78] propose a measure that incorporates both properties simultaneously. In Ref. [77], three new measures were proposed: Simultaneous Diversity and Accuracy (SDAcc), which adapt the measure proposed in Ref. [76] with the aims of correctly classifying all the samples, focusing on the hardest examples; Diversity-Focused-Two (DFTwo) that modifies the SDAcc measure by eliminating those events that take the same decision into account, in order to pay more attention to the ensemble diversity; and Accuracy-Reinforcement (AccRein) that modifies the SDAcc measure by eliminating the event where the candidate model and the current ensemble incorrectly classify the example in order to reinforce the contribution of adding the correct classifier. On the other hand, the work in Ref. [78] takes into account both the margin of examples [79] and the diversity ensemble to assess the importance of individual classifiers. Unlike greedy procedures, a dynamic programming approach that encourages extra diversity was proposed in Ref. [80]. With respect to DES, novel approaches have been proposed that highlight the importance of dynamic selection because a particular subset of the ensemble may not always be optimal for all unknown samples [81–83]. The latest approaches have provided improvements in the strategies to compose the region of competence [74,84,85], reinforcement learning modeling for DES [82] and integration of SES and DES [83], to name just a few.

Although ensemble selection is a very active research topic in the machine learning community [80,83,86], little work has been done on deep CNNs. One can find the SES for visual recognition tasks based on Bayesian Evidence [87] and two DES: 1) a method for medical image classification based on the Dirichlet distribution and the Mahalanobis distance [88], and 2) a method for facial expression recognition based on the geodetic distance calculated from a graph-based pruning approach [86]. On the other hand, one can notice that the pool of candidate models to build an ensemble is usually small in size when deep learning algorithms are used, mainly due to the time required to train them. In this case, when the set is small, greedy selection based on the validation set and a random choice has been shown to provide competitive results [89].

Our research, placed in the SES field, proposes a new metric that takes advantage of the epistemic uncertainty information to select a suitable ensemble subset. Despite the DES approach being interesting, the process involved in selecting the most competitive models for each new sample to be classified implies more demand for computational resources for the solution than SES. The latter is quite relevant in real-time applications where fast and accurate solutions are needed.

3. Forward step-wise Uncertainty-Aware Model Selection

In this section, we explain in depth the steps involved in the proposed Forward Step-wise Uncertainty-Aware Model Selection (FS_UAMS) method of selecting which models to ensemble from a set of deep CNN models.

Let us consider the object recognition problem (e.g. food recognition) with $x^{(i)} \in \mathbb{R}^{wxh}$ inputs features and $y^{(i)} \in \mathbb{R}$ outputs, where i corresponds to the i-th images and wxh are the image dimensions (width and height). As with any supervised machine learning algorithm, in deep CNN the goals are to find the model parameters W of the hypothesis function $h(x)$ that minimize the error between the predicted labels ($y^{(i)}$) and the ground-true labels ($\widehat{y}^{(i)}$). For this purpose, the weights of the deep CNN model (W) are fixed during model training by minimizing the loss function $l(h(x^{(i)}), \widehat{y^{(i)}})$ through a chosen optimization algorithm like: SGD, ADAM, among others.

$$\underset{W}{minimize} \sum_{i=1}^{M} l(h(x^{(i)}), \widehat{y^{(i)}})$$

The results of the optimization after training will often correspond to

a local minimum of the error, mainly because the error surface in the weight space of a deep CNN model is like a landscape with many hills and valleys. Here, the importance of using an ensemble of models is rescued, which smooths the bias produced in the classification by considering the output of several models that converge to different local minima, resulting in better performance (in terms of accuracy) and an improvement in the robustness against noisy inputs.

Particular applications, for example in critical real-time solutions, require accurate and fast models. As well known, by adding more models in an ensemble learning scheme, you can get better accuracy, but worse prediction time. Therefore, it is desirable to find a correct trade-off between both performance metrics. A common strategy for selecting CNN models within a pool of trained models in order to form the ensemble is commonly done by random selection or based on the model's performance in the validation set. Instead, we propose a method that selects and forwards step-wise the models according to the Epistemic Uncertainty analysis.

Let us consider a set $T = \{m_1, m_2, \ldots, m_N\}$ with N trained CNN models and a set $S = \{\varnothing\}$, initially empty, which corresponds to a subset T where N' selected models ($N' < N$) are stored. In the proposed FS_UAMS approach, the Epistemic uncertainty (EU) is analyzed for the purpose of identifying the most confident of the models in its prediction given the input images and, from this information, identifying which models could work best together. Let us define $EU_n \in \mathbb{R}^M$ as EU across all (M) training images for the n-th model as follows:

$$EU_n = \{EU_n^1, \ldots, EU_n^M\},$$

where:

$$EU_n^i = \sum_{y_i} - P_n(y_i|x_i) log(P_n(y_i|x_i))$$

and $P_n(y_i|x_i)$ can be estimated through the MC-dropout method [43], which corresponds to the average of the L softmax outputs obtained from L forward passes, applied to the image x_i during the prediction phase using the model m_n, with dropout turned on. After computing EU_n for all models, we select the first model to form the ensemble according to the minimum mean EU ($\overline{EU_n}$) checking the EU of all models within T. Then, an iterative procedure is carried out to select the following models to form the ensemble based on a proposed criterion that consists of analyzing how different the distribution of the uncertainty is among all images between the selected models ($m_j \in S$) with respect to the candidate model ($m_i \in T$). The candidate model with the smallest difference (minimum distance) is chosen and the procedure starts over until a total of K required models are reached (see Algorithm 3).

Algorithm 1. Forward Step-wise Uncertainty-Aware Model Selection

Algorithm 1: Forward Step-wise Uncertainty-Aware Model Selection

input: Selected models $S \leftarrow \emptyset$, Trained models T, Target models K;
output: Selected models S;
if S *is* $\emptyset$ **then**
  Select the m_i model from T according to the minimum average of epistemic uncertainty across all trained images;
  Update $S \leftarrow S \cup \{m_i\}$;
  Drop m_i from T;
end
while $|S|$ *is less than* K **do**
  $D \leftarrow \emptyset$ // Set that stores the proposed uncertainty-aware distance to select the best model;
  for $m_i \in T$ **do**
    $d_i = 0$;
    for $m_j \in S$ **do**
      $d_i = d_i + ||EU_i - EU_j||_2$
    end
    Update $D \leftarrow D \cup \{d_i\}$;
  end
  Select the m_i model from T according to the minimum value stored in D;
  Update $S \leftarrow S \cup \{m_i\}$;
  Drop m_i from T;
end
return S;

The EU, also known as model uncertainty, is the uncertainty produced by the lack of enough data for model training. For a particular image, the EU can be interpreted as a value that measures the confidence of the model's prediction. The smaller the EU, the greater the confidence for the given prediction and vice versa. In our case we propose to take advantage of this information selecting in the first place the model with greater confidence (smaller uncertainty) and then incorporate more models in the ensemble to slightly disturb the confidence provided for the first models.

Regarding the time complexity of the Algorithm 3, in the worst case (selecting $|T|-1$ models) it has an order of $O(\frac{|T|^3}{6})$ and even less when a small number of candidates are selected (e.g. order $O(\frac{|T|^3}{12})$ selecting $T/2$ models). In practical terms, using a reasonable pool of CNN models, with a size of $|T|=10$, where we intend to select $K=5$ models, the order of complexity of the proposed algorithm tends to be quadratic. Note that the selection process adds an additional cost to the training time to provide the final ensemble model. However, once the models are selected, the proposed method incurs no additional costs at execution time. The latter only depends on the number of selected models and the architecture of each one.

4. Experiments

In this section, we present the datasets used, the evaluation measures, the experimental setup and the results obtained with the proposed FS_UAMS approach in comparison with the baseline selection strategies.

4.1. Datasets for food recognition

In order to evaluate the performance of the proposed method for the food recognition purpose, we selected three challenging food datasets that share the property of having a wide range of dishes, large number of images, and unbalanced data between classes. We describe each one below.

MAFood121 [63] is an international multi-task food image dataset comprising of 21, 175 images belonging to 121 food dishes distributed evenly across 11 most popular types of cuisine. The dataset provides annotations to the images for 3 food-related recognition tasks: a) dish (single-label), b) cuisine (single-label), and c) categories/food groups (multi-label). For training purposes, images are stratified across the dish class, resulting in 72.5% images for training, 12.5% for validation, and the remaining 15% for testing. The images were collected from 4 different sources: 3 public food datasets [8,9,90] and Google Search Engine. As our experiments are focused on food recognition, we only consider the dish annotation for all images.

UECFood256 [9] is a Japanese food images dataset comprising of 31, 395 images belonging to 256 food dishes. In addition to the Japaneses dishes, it also includes traditional dishes from several other countries. The dataset provides two annotations for each image: a) dish and b) dish location (bounding box coordinates). Furthermore, the authors provide an official 5-fold cross-validation split that includes 28, 280 images with a minimum, maximum and average of images per dish class of 87, 637 and 110.47. In our experiments, a Holdout approach was applied, where the first official fold was used for testing purposes and the remainder for training. Regarding the training data, a stratified random sample was applied to split the data into 90% for training and 10% for validation. The bounding box information was not used in our experiments.

VIREO172 [10] is a Chinese multi-task dataset comprising of 110, 241 images belonging to 172 dishes collected by Baidu and Google image search. VIREO Food-172 has annotations for two tasks: a) dish and b) ingredients (only visible). The dataset is distributed as 60% for training, 10% for validation and the remaining 30% for testing. The ingredients annotations were not used in our experiments.

Table 2
Results in terms of accuracy using an ensemble consisting of five ResNet50 models.

Model	MAFood121	UECFood256	VIREO172
FSMS_VAL	85.02%	68.87%	86.35%
FS_UAMS	**85.84%**	**68.94%**	**86.48%**
P(MS-Rand>FS_UAMS)	2.38%	46.83%	7.14%

Table 3
Results in terms of accuracy using an ensemble consisting of five InceptionV3 models.

Model	MAFood121	UECFood256	VIREO172
FSMS_VAL	88.70%	**73.10%**	89.17%
FS_UAMS	**88.95%**	73.01%	**89.26%**
P(MS-Rand>FS_UAMS)	20.24%	25.00%	7.94%

4.2. Validation metrics

Overall accuracy (*Acc*) is the metric selected to evaluate the performance of individual models in the multi-class food recognition problem. Formally, it can be defined as follows:

$$Acc = \frac{1}{M}\sum_{i=1}^{M} ind(h(x^{(i)}), \widehat{y^{(i)}}),$$

where *ind*(*, *) denotes an indicator function that returns 1 when the predicted label ($h(x^{(i)})$) for the image $x^{(i)}$ is equal to its ground-truth label ($\widehat{y^{(i)}}$) and 0, otherwise.

In our case, $h(x^{(i)})$ is computed for the ensemble of deep models by mean of the average of the softmax outputs along all the CNNs that form the ensemble. The equation is defined as follows:

$$h(x^{(i)}) = argmax(\{\overline{softmax(f^{W}(x^{(i)})_1)}, \ldots, \overline{softmax(f^{W^n}(x^{(i)})_C)}\}),$$

with

$$\overline{softmax(f^{W}(x^{(i)})_c)} = \frac{1}{N}\sum_{n=1}^{N} softmax(f^{W^n}(x^{(i)})_c)$$

and

$$softmax(f^{W^n}(x^{(i)})_c) = \frac{\exp(f^{W^n}(x^{(i)})_c)}{\sum_{c'}\exp(f^{W^n}(x^{(i)})_{c'})}$$

where $f^{W^n}(x^{(i)})_c$ corresponds to the final activation of the n-th CNN for the sample $x^{(i)}$ on the c-th label and $c' = 1, \ldots, C$.

4.3. Experimental setup

The proposed FS_UAMS method is evaluated considering two deep CNN architectures: ResNet50 [70] and InceptionV3 [71]. The models were selected taking into account the computational cost ($\leq$200 MB per single sample and model parameters in memory), the training and execution time ($\leq$6 GFLOPS) and the good performance previously shown in food recognition [19,29,39]. Both CNN models were modified by removing the output layer, and instead, we added a dropout layer with a probability of 0.5 and an output layer with softmax activation. The neurons in the output layer equal the number of dishes in the target dataset, specifically 121 for MAFood121, 256 for UECFood256 and 172 for VIREO172.

A total of 10 models was trained for each CNN architecture with the same training hyper-parameters, but randomly changing the weight initialization and shuffling the input images. All of them were re-trained from a pre-trained model on ImageNet [69], during 50 epochs with a

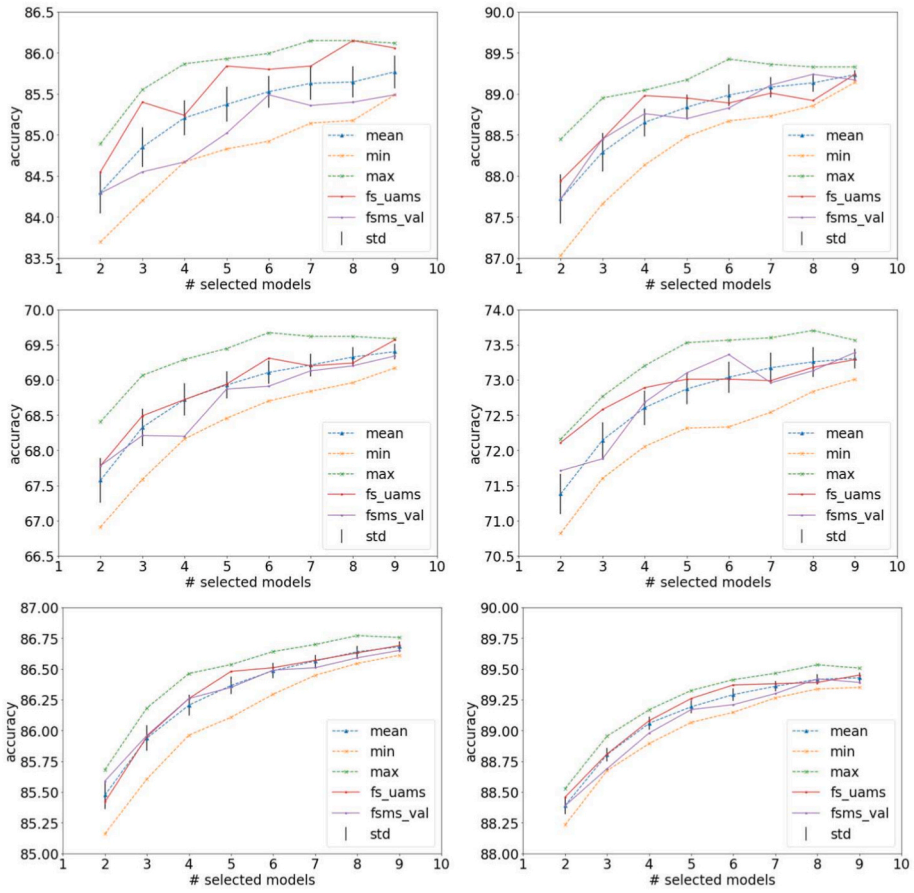

Fig. 2. Performance in terms of accuracy obtained for food recognition using ensemble approaches with different numbers of models. The results on MAFood121, UECFood256 and VIREO172 are shown from the first to the third row, respectively; and the columns group the results with ResNet50 (left) and InceptionV3 (right).

patience of 10, a batch size of 20, an initial learning rate of $2e-4$ and a decay of 0.2 every 8 epochs. As for the data preparation, the images were normalized within the range $[-1.1]$ and the ImageNet mean was subtracted in all of them. Regarding the data augmentation used during the training, the images were first resized to 256x256x3 in the case of ResNet50 and 342x342x3 for InceptionV3. Subsequently, traditional strategies like random crops and horizontal flips were applied. The dimensions of the random crops were 224x224x3 and 299x299x3 for ResNet50 and InceptionV3, respectively. The training was done using categorical cross entropy loss and Adam's optimizer in Keras framework with Tensorflow as backend. The best models of each training were chosen according to the highest accuracy obtained in the validation set.

In addition to the FS_UAMS, two other strategies for model selection were applied for comparative purposes. Selection was done according to the results on the validation sets and randomly selection.

4.4. Results

In order to assess the advantages of the proposed approach with respect to the traditional model selection methods, specifically model selection based on the results of the validation set (FSMS_VAL) or random selection (MS-Rand), ten models are trained for each type of CNN chosen (ResNet50 and InceptionV3) and across three public food datasets. Then, the same trained models have been considered to build the ensembles using the proposed FS_UAMS method or the other methods mentioned above. Note that all reported results refer to the performance of each method evaluated on the test sets. For the UECFood256 case, the validation set is considered as test set. In this way, we compare the performance in the classification of unseen data, which implies comparing the generalization capabilities of each method.

Table 2 and Table 3 summarize the overall accuracy achieved by the ensembles built from each model selection method, with a total of 5 models for each type of CNN architecture, respectively. From the tables, we can see that the proposed method outperforms the results in most cases and in all of them it is a much better option than random selection. Particularly, a significant improvement is provided for FS_UAMS in MAFood121 and VIREO172, where in some cases the probability of obtaining better models is less than 8%. Although it is true that in the case of UECFood256 there is no significant difference between FSMS_VAL with respect to FS_UAMS, our proposal is still better than a random choice.

On the other hand, the performance of each model selection methods (FS_UAMS and FSMS_VAL) and various statistical measures, for all datasets, are shown in Fig. 2. As we can see, ensembles of different size (number of base classifiers) were built and evaluated. Specifically, two to nine different models based on the same CNN architecture were considered for each ensemble. Regarding the statistical measures, the mean, standard deviation (std), minimum (min) and maximum (max) accuracy are presented. The results are distributed for each dataset (rows) and CNN architecture (columns). A noticeable improvement in terms of accuracy can be observed with the proposed FS_UAMS on MAFood121 (first row) and UECFood256 (second row) when we form ensembles with less or equal to five base classifiers, and even more important in some cases our approach achieves almost the maximum performance, which corresponds to the optimal value of the search space when we evaluated all the possible combinations of base classifiers. Regarding the result on VIREO172 (third row), a clear improvement can also be seen with InceptionV3 (right column) mainly with reduced ensemble size. However, with ResNet50 (left column), while it is true that our proposal works better than FSMS_VAL, in most cases the

Table 4
Data showing the percentage of images that change their predicted class positively (N_to_P) and negatively (P_to_N) after adding a new ResNet50 model to the ensemble by mean of the FSMS_VAL approach. *T* denotes the total of images that change their predicted class with respect to the previous ensemble.

	MAFood121			UECFood256			VIREO172		
# Models	T	N_to_P	P_to_N	T	N_to_P	P_to_N	T	N_to_P	P_to_N
2	157	0.6497	0.3503	419	0.6563	0.3437	1363	0.6383	0.3617
3	94	0.6383	0.3617	255	0.5647	0.4353	822	0.5754	0.4246
4	79	0.4810	0.5190	157	0.5287	0.4713	604	0.5828	0.4172
5	55	0.6000	0.4000	139	0.5899	0.4101	478	0.5314	0.4686

Table 5
Data showing the percentage of images that change their predicted class positively (N_to_P) and negatively (P_to_N) after adding a new ResNet50 model to the ensemble by mean of the FS_UAMS approach. *T* denotes the total of images that change their predicted class with respect to the previous ensemble.

	MAFood121			UECFood256			VIREO172		
# Models	T	N_to_P	P_to_N	T	N_to_P	P_to_N	T	N_to_P	P_to_N
2	155	0.7032	0.2968	376	0.6862	0.3138	1388	0.6571	0.3429
3	79	0.6709	0.3291	243	0.5844	0.4156	907	0.5976	0.4024
4	67	0.4627	0.5373	145	0.5448	0.4552	626	0.5799	0.4201
5	65	0.6462	0.3538	115	0.5565	0.4435	490	0.5755	0.4245

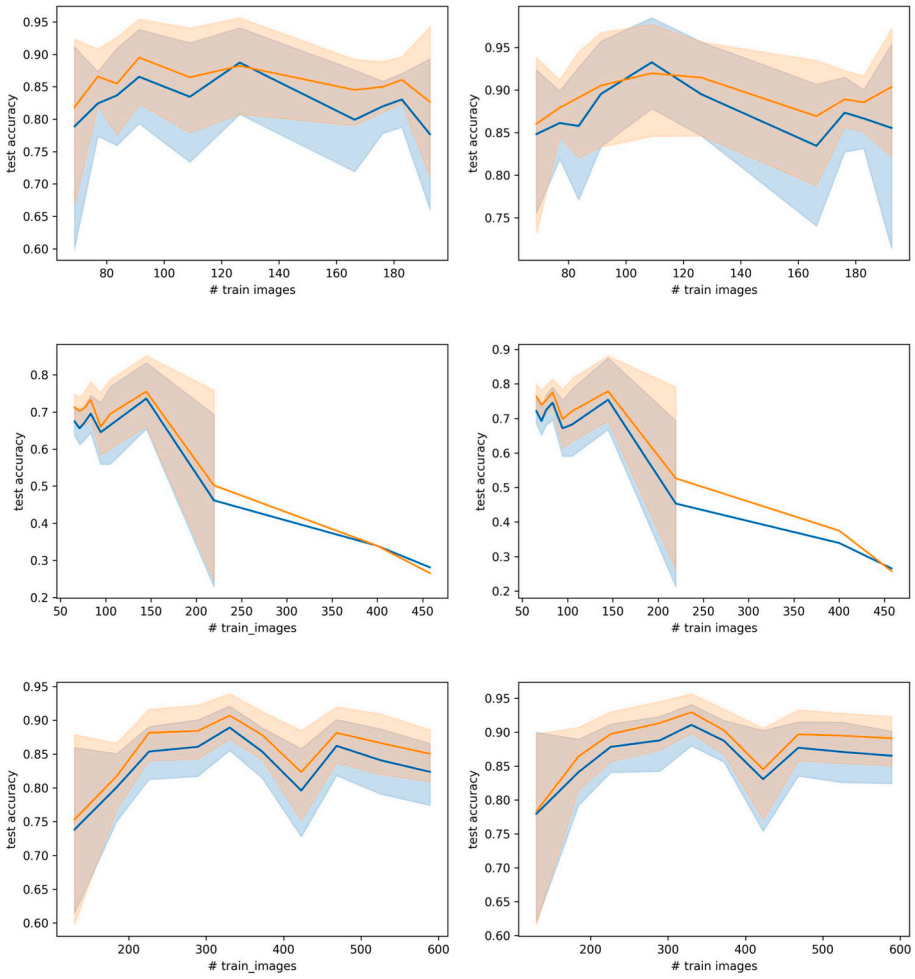

Fig. 3. Plot illustrating the number of training images versus test accuracy achieved. The blue and orange lines correspond to the BC and the ensemble built from FS_UAMS, respectively.

improvement is not too significant. A surprising aspect to highlight with respect to the FSMS_VAL strategy is that in several ensembles built, regardless of the dataset or the CNN architecture, poor performance has been evidenced, at some point close to the minimum value corresponding to the worst combination of the base classifiers. We think the reason for this is that the high dependency on the accuracy in the validation set affects the generalization capability especially when using a dataset with a small or medium number of images per class (<200 on

Table 6
Results for the classes on top-5 accuracy according to the Base Classifier (BC) performance for MAFood121, UECFood256, and VIREO172. The ID column refers to the class identification, BC and Ensemble correspond to the results obtained for BC and the ensemble built from our FS_UAMS approach for the respective classes.

MAFood121			**UECFood256**			**VIREO172**		
ID	BC	Ensemble	ID	BC	Ensemble	ID	BC	Ensemble
59	1.0000	0.9412	174	1.0000	0.9500	93	0.9902	0.9951
2	1.0000	1.0000	158	1.0000	1.0000	165	0.9876	0.9917
54	1.0000	0.9091	243	1.0000	0.9565	113	0.9859	0.9859
4	1.0000	1.0000	27	0.9583	0.9583	98	0.9823	0.9823
116	1.0000	1.0000	77	0.9546	1.0000	171	0.9811	0.9849

Table 7
Results on the classes with the worst-5 accuracy according to the BC performance for MAFood121, UECFood256, and VIREO172. The ID column refers to the class identification, BC and Ensemble correspond to the results obtained for BC and the ensemble built from our FS_UAMS approach for the respective classes.

MAFood121			**UECFood256**			**VIREO172**		
ID	BC	Ensemble	ID	BC	Ensemble	ID	BC	Ensemble
72	0.2778	0.3889	89	0.1429	0.0952	54	0.4685	0.4505
118	0.4546	0.4546	140	0.1500	0.3000	136	0.5172	0.4483
42	0.5455	0.6364	30	0.1905	0.2857	63	0.5298	0.5860
50	0.5484	0.6129	142	0.2105	0.3158	30	0.5730	0.6255
34	0.5714	0.6667	190	0.2222	0.2222	91	0.5859	0.6061

Table 8
MCNemar's test evaluated in ensembles built from FSMS_VAL and FS_UAMS, with respect to the BC and to the ensemble that achieved the minimum performance (Min). For this evaluation, two members are considered to form the ensembles. *** = p-value $\leq$ 0.01, ** = p-value $\leq$ 0.05, * = p-value $\leq$ 0.1 and NS = Not Significant.

	MAFood121		**UECFood256**		**VIREO172**	
Methods	χ^2	p-value	χ^2	p-value	χ^2	p-value
ResNet50						
BC vs FSMS_VAL	5,36	**	19.76	***	61.41	***
BC vs FS_UAMS	14,24	***	11.52	***	45.18	***
Min vs FSMS_VAL	2.27	NS	6.07	**	10.27	***
Min vs FS_UAMS	3.33	*	4.25	**	3.80	*
InceptionV3						
BC vs FSMS_VAL	2.96	*	36.71	***	58.50	***
BC vs FS_UAMS	8.53	***	54.40	***	115.65	***
Min vs FSMS_VAL	3.67	*	4.51	**	1.60	NS
Min vs FS_UAMS	4.10	**	10.03	***	4.63	**

Table 9
Performance of Deep Learning methods for visual food recognition on MAFood121, UECFood256 and VIREO172 datasets.

Method	Accuracy	Ensemble learning	Multiple annotations
MAFood121			
Aguilar et al. [62]	81.62%	no	yes
RUMTL [63]	83.82%	no	yes
TL-Ensemble [39]	84.95%	yes	no
ResNet50	83.16%	no	no
InceptionV3	86.94%	no	no
FS_UAMS	88.95%	yes	no
UECFood256			
DeepFood [16]	54.70%	no	no
Tahir et al. [41]	68.50%	yes	no
WISeR [29]	72.71%	yes	no
ResNet50	66.27%	no	no
InceptionV3	69.72%	no	no
FS_UAMS	73.01%	yes	no
VIREO172			
Arch-D [10]	82.06%	no	yes
RUMTL [63]	85.19%	no	yes
IG-CMAN [34]	90.63%	no	yes
ResNet50	84.45%	no	no
InceptionV3	87.33%	no	no
FS_UAMS	89.26%	yes	no

average). The latter is the case of MAFood121 and UECFood256 datasets. Unlike FSMS_VAL, our proposal based on epistemic uncertainty analysis looks more stable in its performance. In summary, regardless of the number of models used to form the ensemble, in most cases our approach provides better performance than the mean, min, and FSMS_VAL.

In addition to the traditional metric (accuracy) to compare the overall model performance, we analyze if the ensemble of two members built from FSMS_VAL and FS_UAMS strategies provides an improvement statistically significant with respect to the base classifier, which is selected from the best result on test, and the ensemble that provides the minimum accuracy. We validate the statistical significance of the performance improvement using the McNemar's Test [91], which is recommended in cases where it is costly or impractical to use cross-validation [92] (e.g. deep learning models trained on large datasets). For the Test, a 2x2 contingency table was prepared for each pair of methods. On the diagonal, we put the number of miss-classification and correct classification obtained by both methods in the same samples. In the rest, we put the number of miss-classification obtained for one method that are correct for the other in the same sample, and vice versa. The results of the MCNemar's Test can be seen in Table 8, where we observe that there is a statistically significant difference in the performance shown by both strategies compared to the base classifier. It is interesting to note when making the comparison with respect to the ensemble that showed minimum accuracy. In two cases the difference in performance with respect to the ensemble built following the FSMS_VAL strategy does not guarantee to obtain a statistical significance. Summarizing, FS_UAMS provides a statistically significant performance difference in all cases (p-value $\leq$ 0.1), in most of them highly significant (p-value $\leq$ 0.01) and generally a significance level higher than FSMS_VAL.

Table 4 and Table 5 present the total number of images that change the classification results (positively or negatively) after adding a new base classifier to the ensemble by mean of FSMS_VAL or FS_UAMS respectively. When both tables are compared with each other, we can see that in almost all cases the proposed approach tends to maintain the previously well-predicted classes better than FSMS_VAL. That is, with our approach, fewer images are obtained that negatively change their prediction, corresponding to images that are well predicted with the previous ensemble and bad for the current one, which shows that our metric based on epistemic uncertainty analysis is able of finding models

that best complement each other.

The results achieved by the proposed method in each of the classes were also analyzed. Fig. 3 shows the performance achieved for classes with a similar number of training images. To group the classes, a clustering algorithm, *kmeans++* with $k = 10$, was used on the number of training images contained in each class. In this figure, the straight line represents the mean accuracy across all classes within each group, and the background color represents the standard deviation. Regarding the colors, the blue color is used for the BC and the orange color for the proposed Ensemble method. In general terms, the magnitude of improvement when we compare BC with respect to the Ensemble models is almost uniform for classes with different amount of images. Contrary to expectations, the performance improvement of the different classes is not directly proportional to the number of images available in the training sets. Surprisingly, in some cases, classes with more training data perform even worse than classes with fewer samples (see Fig. 3, second row). When analyzing the results obtained in each dataset independently, it is observed that the proposed approach behaves differently for unbalanced data between classes. In MAFood121, a more balanced accuracy can be observed. Regarding UECFood256, a greater increase is observed for classes with few samples and a decrease in performance for classes with large samples. With respect to VIREO172, a greater improvement is observed for the classes with more data. The differences in behavior may be due to the distribution of the data in the training set. Although all datasets used are unbalanced, in some cases there is a strong imbalance in a small number of classes (e.g. UECFood256) or the degree of imbalance between classes is very large (e.g. VIREO172).

Table 6 and Table 7 show the *top-5* and *worst-5* classes predicted for BC and also the performance achieved on these classes with the proposed approach to reflect the improvement obtained in different food classes. In most cases, the performance of the worst classes improves significantly. On the other hand, some higher classes have reduced their performance. From this, it can be inferred that ensemble method tends to balance classification results rather than providing very high performance for a few classes.

Finally, in Table 9, we show the performance achieved on the three public food datasets for the state-of-the-art methods, the base classifier (ResNet50 and InceptionV3), and an ensemble of five InceptionV3 members constructed using the proposed FS_UAMS strategy. In the UECFood256 case, only works that use all uncropped images with the ground-truth bounding box are considered. As expected, our ensemble model provides a much better performance than the base classifier, the improvement is about 2% for MAFood121, 3% for UECFood256 and 2% for VIREO172. Furthermore, we outperform state-of-the-art results in MAFood121 with a wide range and provide competitive results in UECFood256 and VIREO172 using the simplest models to form the ensemble and without requiring additional annotations.

5. Conclusions

In this article, we presented a novel SES method to perform multi-class food recognition problem through an ensemble of Convolutional Neural Networks that employs the Epistemic Uncertainty. Unlike previous ES approaches that use the accuracy or diversity measure to select the base classifiers, in our case we proposed a new measure to select the models based on an Epistemic Uncertainty analysis. The validation of the proposed FS_UAMS ES method was performed on three public food image datasets and was tested using different CNN baseline networks. As a result, we can observe that our approach allowed us to improve the performance in most cases when compared to the baseline ES methods (randomly selection and validation set accuracy). Furthermore, it alleviates the computational cost involved in an exhaustive search of all possible combinations and reduces the bias on the validation set. The proposed method has been validated considering the standard way of generating ensemble in deep learning, that is, combining models belonging to the same CNN architecture. As future work, the compatibility of different architectures to extract the EU will be evaluated to combine with the proposed approach, models based on several architectures. In addition, this research will be extended to develop an uncertainty-aware ensemble selection for multi-label object recognition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by TIN2018-095232-B-C21, SGR-2017 1742, Measurer EIT Digital, Logmeal4Shape and CERCA Programme/Generalitat de Catalunya. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

[1] E. Moguel, J. Berrocal, J. García-Alonso, Systematic literature review of food-intake monitoring in an aging population, Sensors 19 (15) (2019) 3265.
[2] V. Bruno, C.J. Silva Resende, A survey on automated food monitoring and dietary management systems, J. of health & medical informatics 8 (3) (2017).
[3] F. Cordeiro, E. Bales, E. Cherry, J. Fogarty, Rethinking the mobile food journal: exploring opportunities for lightweight photo-based capture, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 3207–3216.
[4] W. Min, S. Jiang, L. Liu, Y. Rui, R. Jain, A survey on food computing, ACM Comput. Surv. 52 (5) (2019) 1–36.
[5] Y. Wang, J.-j. Chen, C.-W. Ngo, T.-S. Chua, W. Zuo, Z. Ming, Mixed dish recognition through multi-label learning, in: Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities, 2019, pp. 1–8.
[6] L. Zhou, C. Zhang, F. Liu, Z. Qiu, Y. He, Application of deep learning in food: a review, Compr. Rev. Food Sci. Food Saf. 18 (6) (2019) 1793–1811.
[7] J. Ahmad, H. Farman, Z. Jan, Deep learning methods and applications, in: Deep Learning: Convergence to Big Data Analytics, Springer, 2019, pp. 31–42.
[8] L. Bossard, M. Guillaumin, L. Van Gool, ECCV, in: Food-101–mining Discriminative Components with Random Forests, Springer, 2014, pp. 446–461.
[9] Y. Kawano, K. Yanai, Automatic expansion of a food image dataset leveraging existing categories with domain adaptation, in: ECCV, Springer, 2014, pp. 3–17.
[10] J. Chen, C.-W. Ngo, Deep-based ingredient recognition for cooking recipe retrieval, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 32–41.
[11] G. Ciocca, P. Napoletano, R. Schettini, Food recognition: a new dataset, experiments, and results, IEEE.J. Biomed. Opt.Health Inf 21 (3) (2016) 588–598.
[12] P. Kaur, K. Sikka, W. Wang, S. Belongie, A. Divakaran, Foodx-251: a Dataset for Fine-Grained Food Classification, 2019 arXiv preprint arXiv:1907.06167.
[13] X. Chen, H. Zhou, Y. Zhu, L. Diao, Chinesefoodnet: A Large-Scale Image Dataset for Chinese Food Recognition, 2017 arXiv preprint arXiv:1705.02743.
[14] S. Mezgec, B.K. Seljak, Using deep learning for food and beverage image recognition, in: 2019 IEEE Int. Conf. On Big Data (Big Data), 2019, pp. 5149–5151. IEEE.
[15] H. Wu, M. Merler, R. Uceda-Sosa, J.R. Smith, Learning to make better mistakes: semantics-aware visual food recognition, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 172–176.
[16] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, Y. Ma, Deepfood: deep learning-based food image recognition for computer-aided dietary assessment, in: International Conference on Smart Homes and Health Telematics, Springer, 2016, pp. 37–48.
[17] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, S. Cagnoni, Food image recognition using very deep convolutional networks, in: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 2016, pp. 41–49.
[18] Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo, T.S. Chua, Food photo recognition for dietary tracking: system and experiment, in: International Conference on Multimedia Modeling, Springer, 2018, pp. 129–141.
[19] E. Tasci, Voting Combinations-Based Ensemble of Fine-Tuned Convolutional Neural Networks for Food Image Recognition, Multimedia Tools and Applications, 2020, pp. 1–22.
[20] P. Pandey, A. Deepthi, B. Mandal, N.B. Puhan, Foodnet: recognizing foods using ensemble of deep networks, IEEE Signal Process. Lett. 24 (12) (2017) 1758–1762.
[21] E. Aguilar, M. Bolaños, P. Radeva, Food recognition using fusion of classifiers based on cnns, in: ICIAP, Springer, 2017, pp. 213–224.
[22] J. Kittler, M. Hatef, R.P. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[23] A. Kendall, Y. Gal, What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? NIPS, 2017, pp. 5574–5584.

[24] Y. Kawano, K. Yanai, Foodcam: a real-time mobile food recognition system employing Fisher vector, in: International Conference on Multimedia Modeling, Springer, 2014, pp. 369–373.
[25] G.M. Farinella, D. Allegra, M. Moltisanti, F. Stanco, S. Battiato, Retrieval and classification of food images, Comput. Biol. Med. 77 (2016) 23–39.
[26] N. Martinel, C. Piciarelli, C. Micheloni, An ensemble feature method for food classification, Mach. Graph. Vis. 26 (2017).
[27] K. Yanai, Y. Kawano, Food image recognition using deep convolutional network with pre-training and fine-tuning, in: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2015, pp. 1–6. IEEE.
[28] P. McAllister, H. Zheng, R. Bond, A. Moorhead, Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets, Comput. Biol. Med. 95 (2018) 217–233.
[29] N. Martinel, G.L. Foresti, C. Micheloni, Wide-slice residual networks for food recognition, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 567–576. IEEE.
[30] K.-H. Lee, X. He, L. Zhang, L. Yang, Cleannet: transfer learning for scalable image classifier training with label noise, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5447–5456.
[31] G.A. Tahir, C.K. Loo, An open-ended continual learning for food recognition using class incremental extreme learning machines, IEEE Access 8 (2020) 82328–82346.
[32] L. Herranz, S. Jiang, R. Xu, Modeling restaurant context for food recognition, IEEE Trans. Multimed. 19 (2) (2016) 430–440.
[33] H. Wang, W. Min, X. Li, S. Jiang, Where and what to eat: Simultaneous restaurant and dish recognition from food image, in: Pacific Rim Conference on Multimedia, Springer, 2016, pp. 520–528.
[34] W. Min, L. Liu, Z. Luo, S. Jiang, Ingredient-guided cascaded multi-attention network for food recognition, in: Proceedings of the 27th ACM International Conference on Multimedia, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1331–1339, https://doi.org/10.1145/3343031.3350948. URL, https://doi.org/10.1145/3343031.3350948.
[35] H. Zhao, K.-H. Yap, A.C. Kot, Fusion learning using semantics and graph convolutional network for visual food recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1711–1720.
[36] E.J. Heravi, H.H. Aghdam, D. Puig, An optimized convolutional neural network with bottleneck and spatial pyramid pooling layers for classification of foods, Pattern Recogn. Lett. 105 (2018) 50–58.
[37] R. Tanno, K. Okamoto, K. Yanai, Deepfoodcam: a dcnn-based real-time mobile food recognition system, in: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 2016, p. 89.
[38] H. Zhao, K.-H. Yap, A.C. Kot, L. Duan, Jdnet: a joint-learning distilled network for mobile visual food recognition, IEEE J. of Selected Topics in Signal Processing 14 (4) (2020) 665–675.
[39] A. Fakhrou, J. Kunhoth, S. Al Maadeed, Smartphone-based food recognition system using multiple deep cnn models, Multimed. Tool. Appl. 80 (21) (2021) 33011–33032.
[40] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conf. On CVPR, 2017, pp. 4700–4708.
[41] G.A. Tahir, C.K. Loo, Explainable deep learning ensemble for food image analysis on edge devices, Comput. Biol. Med. 139 (2021), 104972.
[42] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight Uncertainty in Neural Network, ICML, 2015, pp. 1613–1622.
[43] Y. Gal, Z. Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, ICML, 2016, pp. 1050–1059.
[44] M. Sensoy, L. Kaplan, M. Kandemir, Evidential Deep Learning to Quantify Classification Uncertainty, NIPS, 2018, pp. 3179–3189.
[45] D. Molchanov, A. Ashukha, D. Vetrov, Variational dropout sparsifies deep neural networks, in: ICML-volume 70, JMLR. org, 2017, pp. 2498–2507.
[46] C. Louizos, M. Welling, Multiplicative normalizing flows for variational bayesian neural networks, in: ICML-volume 70, JMLR. org, 2017, pp. 2218–2227.
[47] P. Van Molle, T. Verbelen, B. Vankeirsbilck, J. De Vylder, B. Diricx, T. Kimpe, P. Simoens, B. Dhoedt, Leveraging the bhattacharyya coefficient for uncertainty quantification in deep neural networks, Neural Comput. Appl. (2021) 1–17.
[48] Y. Gal, R. Islam, Z. Ghahramani, Deep bayesian active learning with image data, in: Proc. Of the 34th ICML-Volume 70, JMLR. org, 2017, pp. 1183–1192.
[49] S. Khan, M. Hayat, S.W. Zamir, J. Shen, L. Shao, Striking the right balance with uncertainty, in: Proc. Of the IEEE Conf. on CVPR, 2019, pp. 103–112.
[50] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, in: ICML, vol. 2, 2016, p. 7.
[51] J. Chang, Z. Lan, C. Cheng, Y. Wei, Data uncertainty learning in face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5710–5719.
[52] M. Cai, F. Lu, Y. Sato, Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14392–14401.
[53] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, H. Roth, 3d semi-supervised learning with uncertainty-aware multi-view co-training, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3646–3655.
[54] C. Sakaridis, D. Dai, L.V. Gool, Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7374–7383.
[55] J. Choi, D. Chun, H. Kim, H.-J. Lee, Gaussian yolov3: an accurate and fast object detector using localization uncertainty for autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 502–511.
[56] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2888–2897.
[57] L. Bertoni, S. Kreiss, A. Alahi, Monoloco: monocular 3d pedestrian localization and uncertainty estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6861–6871.
[58] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F.S. Saleh, T. Zhang, N. Barnes, Uc-net, Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8582–8591.
[59] A. Kumar, T.K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, C. Feng, Luvli face alignment: estimating landmarks' location, uncertainty, and visibility likelihood, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8236–8246.
[60] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, J. Schneider, Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2095–2104.
[61] Z. Zhang, A. Romero, M.J. Muckley, P. Vincent, L. Yang, M. Drozdzal, Reducing uncertainty in undersampled mri reconstruction with active acquisition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2049–2058.
[62] E. Aguilar, P. Radeva, Uncertainty-aware integration of local and flat classifiers for food recognition, Pattern Recogn. Lett. 136 (2020) 237–243.
[63] E. Aguilar, M. Bolaños, P. Radeva, Regularized uncertainty-based multi-task learning model for food analysis, J. Vis. Commun. Image Represent. 60 (2019) 360–370.
[64] E. Aguilar, B. Nagarajan, R. Khatun, M. Bolaños, P. Radeva, Uncertainty Modeling and Deep Learning Applied to Food Image Analysis, BIODEVICES, 2020, pp. 9–16.
[65] G. Huang, Y. Li, G. Pleiss, Z. Liu, J.E. Hopcroft, K.Q. Weinberger, Snapshot ensembles: train 1, get M for free, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.
[66] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[67] R.E. Schapire, The Boosting Approach to Machine Learning: an Overview, Nonlinear Estimation and Classification, 2003, pp. 149–171.
[68] D.H. Wolpert, Stacked generalization, Neural Network. 5 (2) (1992) 241–259.
[69] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, NIPS, 2012, pp. 1097–1105.
[70] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. Of the IEEE Conf. on CVPR, 2016, pp. 770–778.
[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conf. On CVPR, 2016, pp. 2818–2826.
[72] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.
[73] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, p. 18.
[74] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, F. Herrera, Dynamic ensemble selection for multi-class imbalanced datasets, Inf. Sci. 445 (2018) 22–37.
[75] I. Partalas, G. Tsoumakas, I.P. Vlahavas, Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection, ECAI, 2008, pp. 117–121.
[76] I. Partalas, G. Tsoumakas, I. Vlahavas, An ensemble uncertainty aware measure for directed hill climbing ensemble pruning, Mach. Learn. 81 (3) (2010) 257–282.
[77] Q. Dai, R. Ye, Z. Liu, Considering diversity and accuracy simultaneously for ensemble pruning, Appl. Soft Comput. 58 (2017) 75–91.
[78] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, M. Xu, Margin & diversity based ordering ensemble pruning, Neurocomputing 275 (2018) 237–246.
[79] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, et al., Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[80] O.A. Alzubi, J.A. Alzubi, M. Alweshah, I. Qiqieh, S. Al-Shami, M. Ramachandran, An optimal pruning algorithm of classifier ensembles: dynamic programming approach, Neural Comput. Appl. 32 (20) (2020) 16091–16107.
[81] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Dynamic classifier selection: recent advances and perspectives, Inf. Fusion 41 (2018) 195–216.
[82] Z. Liu, K. Ramamohanarao, Instance-based ensemble selection using deep reinforcement learning, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–7. IEEE.
[83] T.T. Nguyen, A.V. Luong, M.T. Dang, A.W.-C. Liew, J. McCall, Ensemble selection based on classifier prediction confidence, Pattern Recogn. 100 (2020), 107104.
[84] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Prototype selection for dynamic classifier and ensemble selection, Neural Comput. Appl. 29 (2) (2018) 447–457.
[85] D.V. Oliveira, G.D. Cavalcanti, R. Sabourin, Online pruning of base classifiers for dynamic ensemble selection, Pattern Recogn. 72 (2017) 44–58.
[86] D. Li, G. Wen, X. Li, X. Cai, Graph-based dynamic ensemble pruning for facial expression recognition, Appl. Intell. 49 (9) (2019) 3188–3206.
[87] Y.-D. Kim, T. Jang, B. Han, S. Choi, Learning to select pre-trained deep representations with bayesian evidence framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5318–5326.

[88] A.G. Pacheco, T. Trappenberg, R.A. Krohling, Learning dynamic weights for an ensemble of deep models applied to medical imaging classification, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.
[89] F. Perez, S. Avila, E. Valle, Solo or ensemble? choosing a cnn architecture for melanoma classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, 0–0.
[90] C. Güngör, F. Baltacı, A. Erdem, E. Erdem, Turkish cuisine: a benchmark dataset with Turkish meals for food recognition, in: 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, pp. 1–4, https://doi.org/10.1109/SIU.2017.7960494.
[91] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika 12 (2) (1947) 153–157.
[92] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923.