Food Recognition Using Fusion of Classifiers Based on CNNs

Eduardo Aguilar(✉), Marc Bolaños, and Petia Radeva

Universitat de Barcelona & Computer Vision Center, Barcelona, Spain
{eduardo.aguilar,marc.bolanos,petia.ivanova}@ub.edu

Abstract. With the arrival of Convolutional Neural Networks, the complex problem of food recognition has experienced an important improvement recently. The best results have been obtained using methods based on very deep Convolutional Neural Networks, which show that the deeper the model, the better the classification accuracy is. However, very deep neural networks may suffer from the overfitting problem. In this paper, we propose a combination of multiple classifiers based on Convolutional models that complement each other and thus, achieve an improvement in performance. The evaluation of our approach is done on 2 public datasets: Food-101 as a dataset with a wide variety of fine-grained dishes, and Food-11 as a dataset of high-level food categories, where our approach outperforms the independent Convolutional Neural Networks models.

Keywords: Food recognition · Fusion classifiers · CNN

1 Introduction

In the field of computer vision, food recognition has caused a lot of interest for researchers considering its applicability in solutions that improve people's nutrition and hence, their lifestyle [1]. In relation to the healthy diet, traditional strategies for analyzing food consumption are based on self-reporting and manual quantification [2]. Hence, the information used to be inaccurate and incomplete [3]. Having an automatic monitoring system and being able to control the food consumption is of vital importance, especially for the treatment of individuals who have eating disorders, want to improve their diet or reduce their weight.

Food recognition is a key element within a food consumption monitoring system. Originally, it has been approached by using traditional approaches [4,5], which extracted ad-hoc image features by means of algorithms based mainly on color, texture and shape. More recently, other approaches focused on using Deep Learning techniques [5–8]. In these works, feature extraction algorithms are not hand-crafted and additionally, the models automatically learn the best way to discriminate the different classes to be classified. As for the results obtained, there is a great difference (more than 30%) between the best method based on hand-crafted features compared to newer methods based on Deep Learning, where the best results have been obtained with Convolutional Neural Networks (CNN) architectures that used inception modules [8] or residual networks [7].

S. Battiato et al. (Eds.): ICIAP 2017, Part II, LNCS 10485, pp. 213–224, 2017.
https://doi.org/10.1007/978-3-319-68548-9_20

Fig. 1. Example images of Food-101 dataset. Each image represents a dish class.

Food recognition can be considered as a special case of object recognition, being a very active topic in computer vision lately. The specific part is that dish classes have a much higher inter-class similarity and intra-class variation than usual Imagenet objects (cars, animals, rigid objects, etc.) (see Fig. 1). If we analyze the last accuracy increase in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9], it has been improved thanks to the depth increase of CNN models [10–13] and also to the fusion of CNNs models [11,13]. The main problem of CNNs is the need of large datasets to avoid overfitting the network as well as the need of high computational power for training them.

Considering the use of different classifiers, in general, trained on the same data, one can observe that patterns misclassified by the different models would not necessarily overlap [14]. This suggests that they could potentially offer complementary information that can be used to improve the final performance [14]. An option to combine the outputs of different classifiers was proposed in [15], where the authors used what they call a decision templates scheme instead of simple aggregation operators such as the product or average. As they showed, this scheme maintains a good performance using different training set sizes and is also less sensitive to particular datasets compared to the other schemes.

In this article, we integrate the fusion concept into the CNN framework, with the purpose of demonstrating that the combination of the classifiers' output, by using a decision template scheme, allows to improve the performance on the food recognition problem. Our contributions are the following: (1) we propose the first food recognition algorithm that fuses the output of different CNN models, (2) we show that our CNNs fusion approach has better performance compared to the use of CNN models separately, and (3) we demonstrate that our CNNs Fusion approach keeps a high performance independently of the target (dishes, family of dishes) and dataset validating it on 2 public datasets.

The organization of the article is as follows. In Sect. 2, we present the CNNs Fusion methodology. In Sect. 3, we present the datasets, the experimental setup and discuss the results. Finally, in Sect. 4, we describe the conclusions.

2 Methodology

In this section, we describe the CNN Fusion methodology (see Fig. 2), which is composed of two main steps: training K CNN models based on different architectures and fusing the CNN outputs using the decision templates scheme.

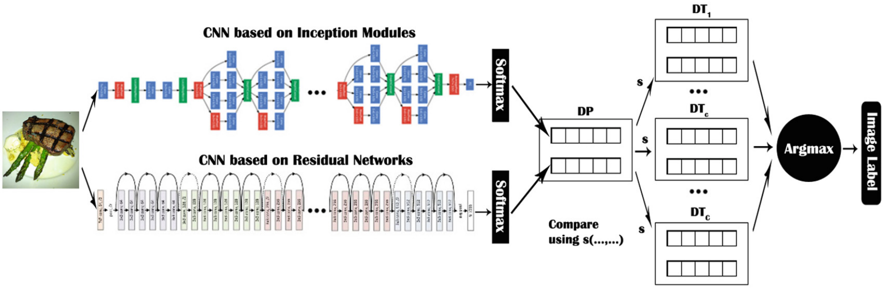

Fig. 2. General scheme of our CNNs fusion approach.

2.1 Training of CNN Models

The first step in our methodology involves separately training two CNN models. We chose two different kind of models winners of the ILSVRC in the object recognition task. Both models won or are based on the winner of the challenges made in 2014 and 2015 proposing novel architectures: the first based its design on “inception models” and the second on “residual networks”. First, each model was pre-trained on the ILSVRC data. Later, all layers were fine-tuned by a certain number of epochs, selecting for each one the model that provides the best results in the validation set and that will be used in the fusion step.

2.2 Decision Templates for Classifiers Fusion

Once we trained the models on the food dataset, we combined the softmax classifier outputs of each model using the Decision Template (DT) scheme [15].

Let us annotate the output of the last layer of the k-th CNN model as $(\omega_{1,k}, \dots, \omega_{C,k})$, where $c = 1, \dots, C$ is the number of classes and $k = 1, \dots K$ is the index of the CNN model (in our case, K= 2). Usually, the softmax function is applied, to obtain the probability value of model k to classify image x to a class c: $p_{k,c}(x) = \frac{e^{\omega_{k,c}}}{\sum_{c=1}^{C} e^{\omega_{k,c}}}$. Let us consider the k-th decision vector D_k:

$$D_k(x) = [p_{k,1}(x), p_{k,2}(x), \dots, p_{k,C}(x)]$$

Definition [15]**:** A **Decision Profile,** DP for a given image x is defined as:

$$DP(x) = \begin{bmatrix} p_{1,1}(x) \; p_{1,2}(x) \; \dots \; p_{1,C}(x) \\ \dots \\ p_{K,1}(x) \; p_{K,2}(x) \; \dots \; p_{K,C}(x) \end{bmatrix} \tag{1}$$

Definition [15]**:** Given N training images, a **Decision Template** is defined as a set of matrices $DT = (DT^1, \dots, DT^C)$, where the c-th element is obtained as the average of the decision profiles (1) on the training images of class c:

$$DT^c = \frac{\sum_{j=1}^{N} DP(x_j) \times Ind(x_j, c)}{\sum_{j=1}^{N} Ind(x_j, c)},$$

where $Ind(x_j, c)$ is an indicator function with value 1 if the training image x_j has a crisp label c, and 0, otherwise [16].

Finally, the resulting prediction for each image is determined considering the similarity $s(DP(x), DT^c(x))$ between the decision profile $DP(x)$ of the test image and the decision template of class c,$c = 1, \ldots, C$. Regarding the arguments of the similarity function $s(.,.)$ as fuzzy sets on some universal set with $K \times C$ elements, various fuzzy measures of similarity can be used. We chose different measures [15], namely 2 measures of similarity, 2 inclusion indices, a consistency measure and the Euclidean Distance. These measures are formally defined as:

$$S_1(DT^c, DP(x)) = \frac{\sum_{k=1}^{K}\sum_{i=1}^{C} \min(DT^c_{k,i}, DP_{k,i}(x))}{\sum_{k=1}^{K}\sum_{i=1}^{C} \max(DT^c_{k,i}, DP_{k,i}(x))},$$

$$S_2(DT^c, DP(x)) = 1 - \sup_u\{\left|DT^c_{k,i} - DP_{k,i}(x)\right| : c = 1, \ldots, C, k = 1, \ldots, K\},$$

$$I_1(DT^c, DP(x)) = \frac{\sum_{k=1}^{K}\sum_{i=1}^{C} \min(DT^c_{k,i}, DP_{k,i}(x))}{\sum_{k=1}^{K}\sum_{i=1}^{C} DT^c_{k,i}},$$

$$I_2(DT^c, DP(x)) = \inf_u\{\max(\overline{DT^c_{k,i}}, DP_{k,i}(x)) : c = 1, \ldots, C, k = 1, \ldots, K\},$$

$$C(DT^c, DP(x)) = \sup_u\{\min(DT^c_{k,i}, DP_{k,i}(x)) : c = 1, \ldots, C, k = 1, \ldots, K\},$$

$$N(DT^c, DP(x)) = 1 - \frac{\sum_{k=1}^{K}\sum_{i=1}^{C}(DT^c_{k,i} - DP_{k,i}(x))^2}{K \times C},$$

where $DT^c_{k,i}$ is the probability assigned to the class i by the classifier k in the DT^c, $\overline{DT^c_{k,i}}$ is the complement of $DT^c_{k,i}$ calculated as $1 - DT^c_{k,i}$, and $DP_{k,i}(x)$ is the probability assigned by the classifier k to the class i in the DP calculated for the image, x. The final label, L is obtained as the class that maximizes the similarity, s, the inclusion index, the consistency measure or the Euclidean distance between $DP(x)$ and DT^c: $L(x) = argmax_{c=1,\ldots,C}\{s(DT^c, DP(x))\}$.

3 Experiments

3.1 Datasets

The data used to evaluate our approach are two public datasets of very different images: Food-11 [17] and Food-101 [4], which are chosen in order to verify that the classifiers fusion provides good results regardless of the different properties of the target datasets, such as intra-class variability (the first one is composed of many dishes of the same general category, while the second one is composed of specific fine-grained dishes), inter-class similarity, number of images, number of classes, images acquisition condition, among others.

Food-11 is a dataset for food recognition [17], which contains 16,643 images grouped into 11 general categories of food: bread, dairy products, dessert, egg, fried food, meat, noodle/pasta, rice, seafood, soup and vegetable/fruit (see Fig. 3). The images were collected from existing food datasets (Food-101, UECFOOD100, UECFOOD256) and social networks (Flickr, Instagram). This dataset has an unbalanced number of images for each class with an average of 1,513 images per class and a standard deviation of 702. For our experiments, we used the same data split, images and proportions, provided by the authors [17]. These are divided as 60% for training, 20% for validation and 20% for test, that is 9,866, 3,430 and 3,347 images for each set, respectively.

Fig. 3. Images from the Food-11 dataset. Each image corresponds to a different class.

Food-101 is a standard to evaluate the performance of visual food recognition [4]. This dataset contains 101.000 real-world food images downloaded from foodspotting.com, which were taken under unconstrained conditions. The authors chose the top 101 most popular classes of food (see Fig. 1) and collected 1,000 images for each class: 75% for training and 25% for testing. With respect to the classes, these consist of very diverse and fine-grained dishes of various countries, but also with highly intra-class variation and inter-class similarity in most occasions. In our experiments, we used the same data splits provided by the authors. Unlike Food-11, and keeping the procedure followed by other authors [5,7,8], we validate and test our model on the same data split.

3.2 Experimental Setup

As usually, every CNN model was pre-trained on the ILSVRC dataset. Following, we adapted them by changing the output of the models to the number of classes for each target dataset and fine-tuned the models using the new images. For the training of the CNN models, we used the Deep Learning framework Keras[1]. The models chosen for Food-101 dataset due to their performance-efficiency ratio were InceptionV3 [18] and ResNet50 [13]. Both models were trained during 48 epochs with a batch size of 32, and a learning rate of 5×10^{-3} and 1×10^{-3}, respectively. In addition, we applied a decay of 0.1 during the training of InceptionV3 and of 0.8 for ResNet50 every 8 epochs. The parameters were chosen empirically by analyzing the training loss.

As to the Food-11 dataset, we kept the ResNet50 model, but changed InceptionV3 by GoogLeNet [12], since InceptionV3 did not generalize well over Food-11. We believe that the reason is the small number of images for each class not sufficient to avoid over-fitting; the model quickly obtained a good result on the training set, but a poor performance on the validation set. GoogLeNet and Resnet50 were trained during 32 epochs with a batch size of 32 and 16,

[1] www.keras.io.

respectively. The other parameters used for the ResNet50 were the same used for Food-101. In the case of GoogLeNet, we used a learning rate of 1×10^{-3} and applied a decay of 0.1 during every 8 epochs, that turned out empirically the optimal parameters for our problem.

3.3 Data Preprocessing and Metrics

The preprocessing made during the training, validation and testing phases was the following. During the training of our CNN models, we applied different preprocessing techniques on the images with the aim of increasing the samples and to prevent the over-fitting of the networks. First, we resized the images keeping the original aspect ratio as well as satisfying the following criteria: the smallest side of the resulting images should be greater than or equal to the input size of the model; and the biggest side should be less than or equal to the maximal size defined in each model to make random crops on the image. In the case of InceptionV3, we set to 320 pixels as maximal size, for GoogLeNet and ResNet50 the maximal size was defined as 256 pixels. After resizing the images, inspired by [8], we enhanced them by means of a series of random distortions such as: adjusting color balance, contrast, brightness and sharpness. Finally, we made random crops of the images, with a dimension of 299 × 299 for InceptionV3 and of 224 × 224 for the other models. Then, we applied random horizontal flips with a probability of 50%, and subtracted the average image value of the ImageNet dataset. During validation, we applied a similar preprocessing, with the difference that we made a center crop instead of random crops and that we did not apply random horizontal flips. During test, we followed the same procedure than in validation (1-Crop evaluation). Furthermore, we also evaluated the CNN using 10-Crops, which are: upper left, upper right, lower left, lower right and center crop, both in their original setup and also applying an horizontal flip [10]. As for 10-Crops evaluation, the classifier gets a tentative label for each crop, and then majority voting is used over all predictions. In the cases where two labels are predicted the same number of times, the final label is assigned comparing their highest average prediction probability.

We used four metrics to evaluate the performance of our approach, overall Accuracy (ACC), Precision (P), Recall (R), and F_1 score.

3.4 Experimental Results on Food-11

The results obtained during the experimentation on Food-11 dataset are shown in Table 1 giving the error rate (1 - accuracy) for the best CNN models, compared to the CNNs Fusion. We report the overall accuracy by processing the test data using two procedures: (1) a center crop (1-Crop), and (2) using 10 different crops of the image (10-Crops). The experimental results show an error rate of less than 10 % for all classifiers, achieving a slightly better performance when using 10-Crops. The best accuracy is achieved with our CNNs Fusion approach, which is about 0.75% better than the best result of the classifiers evaluated separately. On the other hand, the baseline classification on Food-11 was given by their authors, who obtained an overall accuracy of 83.5% using GoogLeNet models fine-tuned in the last six layers without any pre-processing and post-processing steps. Note that the best results obtained with our approach have been using the pointwise measures (S2, I2). The particularity of these measures is that they penalize big differences between corresponding values of DTs and DP being from the specific class to be assigned as the rest of the class values. From now on, in this section we only report the results based on the 10-Crops procedure.

Table 1. Overall test set error rate of Food-11 obtained for each model. The distance measure is shown between parenthesis in the CNNs Fusion models.

Authors	Model	1-Crop	10-Crops	N/A
[17]	GoogLeNet	-	-	16.5%
us	GoogLeNet	9.89%	9.29%	-
us	ResNet50	6.57%	6.39%	-
us	CNNs Fusion (S_1)	6.36%	5.86%	-
us	CNNs Fusion (S_2)	6.12%	**5.65%**	-
us	CNNs Fusion (I_1)	6.36%	5.89%	-
us	CNNs Fusion (I_2)	6.30%	**5.65%**	-
us	CNNs Fusion (C)	6.45%	6.07%	-
us	CNNs Fusion (N)	6.36%	5.92%	-

As shown in Table 2, the CNNs Fusion is able to properly classify not only the images that were correctly classified by both baselines, but in some occasions also when one or both fail. This suggests that in some cases both classifiers may be close to predicting the correct class and combining their outputs can make a better decision.

Table 2. Percentage of images well-classified and misclassified on Food-11 using our CNNs Fusion approach, distributed by the results obtained with GoogLeNet (CNN_1) and ResNet50 (CNN_2) models independently evaluated.

CNNs Fusion	CNNs evaluated independently			
	Both wrong	CNN_1 wrong	CNN_2 wrong	Both fine
Well-classified	3.08%	81.77%	54.76%	99.97%
Misclassified	96.92%	18.23%	45.24%	0.03%

Samples misclassified by our model are shown in Fig. 4, where most of them are produced by mixed items, high inter-class similarity and wrongly labeled images. We show the ground truth (top) and the predicted class (bottom) for each sample image.

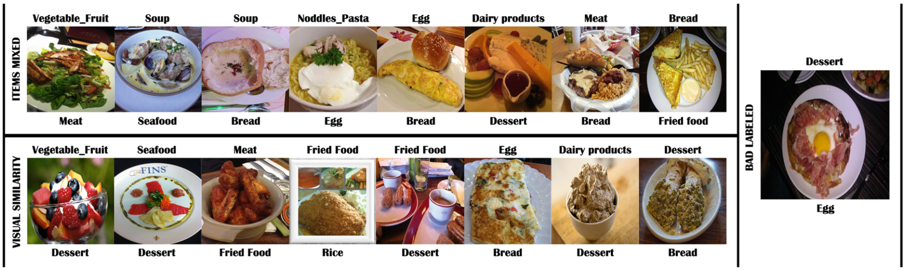

Fig. 4. Misclassified Food-11 examples: predicted labels (on the top), and the groundtruth (on the bottom).

In Table 3, we show the precision, recall and F_1 score obtained for each class separately. By comparing the F_1 score, the best performance is achieved for the class Noodles_Pasta and the worst for Dairy products. Specifically, the class Noddles_Pasta only has one image misclassified, which furthermore is a hard sample, because it contains two classes together (see items mixed in Fig. 4). Considering the precision, the worst results are obtained for the class Bread, which is understandable considering that bread can sometimes be present in other classes (e.g. soup or egg). In the case of recall, the worst results are obtained for Dairy products, where an error greater than 8% is produced for misclassifying several images as class Dessert. The cause of this is mainly, because the class Dessert has a lot of items in their images that could also belong to the class Dairy products (e.g. frozen yogurt or ice cream) or that are visually similar.

Table 3. Some results obtained on the Food-11 using our CNNs Fusion approach.

Class	#Images	Precision	Recall	F1
Bread	368	**88.95%**	91.85%	90.37%
Dairy products	148	89.86%	**83.78%**	**86.71%**
Meat	432	94.12%	92.59%	93.35%
Noodles_Pasta	147	**100.00%**	**99.32%**	**99.66%**
Rice	96	94.95%	97.92%	96.41%
Vegetable_Fruit	231	98.22%	95.67%	96.93%

3.5 Experimental Results on Food-101

The overall accuracy on Food-101 dataset is shown in Table 4 for two classifiers based on CNN models, and also for our CNNs Fusion. The overall accuracy is obtained by means of the evaluation of the prediction using 1-Crop and 10-Crops. The experimental results show better performance (about 1% more) using 10-Crops instead of 1-Crop. From now on, in this section we only report the results based on the 10-Crops procedure. In the same way as observed in Food-11, the best accuracy obtained with our approach was by means of pointwise measures S2, I2, where the latter provides a slightly better performance. Again, the best accuracy is also achieved by the CNNs Fusion, which is about 1.5% higher than the best result of the classifiers evaluated separately. Note that the best performance on Food-101 (overall accuracy of 90.27%) was obtained using WISeR [7]. In addition, the authors show the performance by another deep learning-based approaches, in which three CNN models achieved over a 88% (InceptionV3, ResNet200 and WRN [19]). However, WISeR, WRN and ResNet200 models were not considered in our experiments since they need a multi-GPU server to replicate their results. In addition, those models have 2.5 times more parameters than the models chosen, which involve a high cost computational especially during the learning stage. Following the article steps, our best results replicating the methods were those using InceptionV3 and ResNet50 models used as a base to evaluate the performance of our CNNs Fusion approach.

Table 4. Overall test set accuracy of Food-101 obtained for each model.

Author	Model	1-Crop	10-Crops	N/A
[8]	InceptionV3	-	-	88.28%
[7]	ResNet200	-	88.38%	-
[7]	WRN	-	88.72%	-
[7]	WISeR	-	90.27%	-
us	ResNet50	82.31%	83.54%	-
us	InceptionV3	83.82%	84.98%	-
us	CNNs Fusion (S_1)	85.52%	86.51%	-
us	CNNs Fusion (S_2)	86.07%	86.70%	-
us	CNNs Fusion (I_1)	85.52%	86.51%	-
us	CNNs Fusion (I_2)	85.98%	**86.71**%	-
us	CNNs Fusion (C)	85.24%	86.09%	-
us	CNNs Fusion (N)	85.53%	86.50%	-

As shown in Table 5, in this dataset the CNNs Fusion is also able to properly classify not only the images that were correctly classified for both classifiers, but also when one or both fail. Therefore, we demonstrate that our proposed approach maintains its behavior independently of the target dataset.

Table 6 shows the top five worst and best classification results on Food-101 classes. We highlight the classes with the worst and best results. As for the worst class (Steak), the precision and recall achieved are 60.32% and 59.60%, respectively. Interestingly, about 26% error in the precision and 30% error in the recall is produced with only three classes: Filet mignon, Pork chop and Prime rib. As shown in Fig. 5, these are fine-grained classes with high inter-class similarities

Table 5. Percentage of images well-classified and misclassified on Food-101 using our CNNs Fusion approach, distributed by the results obtained with InceptionV3 (CNN_1) and ResNet50 (CNN_2) models independently evaluated.

CNNs Fusion	CNNs evaluated independently			
	Both wrong	CNN_1wrong	CNN_2wrong	Both fine
Well-classified	1.95%	73.07%	64.95%	99.97%
Misclassified	98.05%	26.93%	35.05%	0.03%

Table 6. Top 3 better and worst classification results on Food-101.

Class	Precision	Recall	F1
Spaghetti Bolognese	94.47%	95.60%	95.03%
Macarons	97.15%	95.60%	96.37%
Edamame	**99.60%**	**100.00%**	**99.80%**
Steak	**60.32%**	**59.60%**	**59.96%**
Pork Chop	75.71%	63.60%	69.13%
Foie Gras	72.96%	68.00%	70.39%

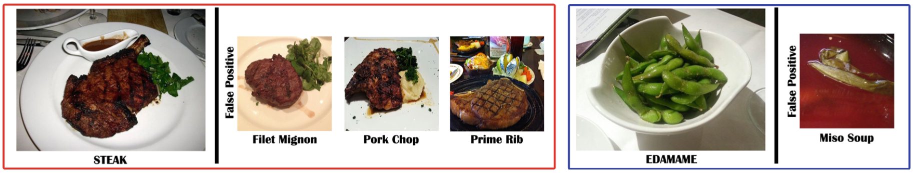

Fig. 5. Misclassified examples for the Food-101 classes that obtained the worst (steak) and best (edamame) classification results by F1 score (groundtruth label - bottom).

that imply high difficulty for the classifier, because it should identify small details that allow to determine the corresponding class of the images. On the other hand, the best class (Edamame) was classified achieving 99.60% of precision and 100% of recall. Unlike Steak, Edamame is a simple class to classify, because it has a low intra-class variation and low inter-class similarities. In other words, the images in this class have a similar visual appearance and they are quite different from the images of the other classes. Regarding the only one misclassified image, its visual appearance is close to the class Edamame as for the shape and color.

4 Conclusions

In this paper, we addressed the problem of food recognition and proposed a CNNs Fusion approach based on the concepts of decision templates and decision profiles and their similarity that improves the classification performance with respect to using CNN models separately. Evaluating different similarity measures, we show that the optimal one is based on the infinimum of the maximum between the complementary of the decision templates and the decision profile of the test images. On Food-11, our approach outperforms the baseline accuracy by more than 10% of accuracy. As for Food-101, we used two CNN architectures providing the best state of the art results where our CNNs Fusion strategy outperformed them again. As a future work, we plan to evaluate the performance of the CNN Fusion strategy as a function of the number of CNN models.

Acknowledgement. This work was partially funded by TIN2015-66951-C2, SGR 1219, CERCA, *ICREA Academia'2014*, CONICYT Becas Chile, FPU15/01347 and Grant 20141510 (Marató TV3). The funders had no role in the study design, data collection, analysis, and preparation of the manuscript. We acknowledge Nvidia Corporation for the donation of 3 Titan X GPUs.

References

1. Waxman, A., Norum, K.R.: WHO global strategy on diet, physical activity and health. Food Nutr. Bull. **25**, 292–302 (2004)
2. Shim, J.-S., Oh, K., Kim, H.C.: Dietary assessment methods in epidemiologic studies. Epidemiol. Health **36**, e2014009 (2014)
3. Rumpler, W.V., Kramer, M., Rhodes, D.G., Moshfegh, A.J., Paul, D.R.: Identifying sources of reporting error using measured food intake. Eur. J. Clin. Nutr. **62**, 544–552 (2008)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101-mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 446–461. Springer, Cham (2014). doi:10.1007/978-3-319-10599-4_29
5. Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Ma, Y.: DeepFood: deep learning-based food image recognition for computer-aided dietary assessment. In: Chang, C.K., Chiari, L., Cao, Y., Jin, H., Mokhtari, M., Aloulou, H. (eds.) ICOST 2016. LNCS, vol. 9677, pp. 37–48. Springer, Cham (2016). doi:10.1007/978-3-319-39601-9_4
6. Yanai, K., Kawano, Y.: Food image recognition using deep convolutional network with pre-training and fine-tuning. In: ICMEW, pp. 1–6 (2015)
7. Martinel, N., Foresti, G.L., Micheloni, C.: Wide-slice residual networks for food recognition. arXiv Preprint (2016)
8. Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., Cagnoni, S.: Food image recognition using very deep convolutional networks. In: Proceedings of the 2nd International Workshop on MADiMa, pp. 41–49 (2016)
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1–9 (2012)
11. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). doi:10.1007/978-3-319-10590-1_53

12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR, pp. 1–9 (2015)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
14. Kittler, J., Hatef, M.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 226–239 (1998)
15. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recogn. **34**, 299–314 (2001)
16. Kuncheva, L.I., Kounchev, R.K., Zlatev, R.Z.: Aggregation of multiple classification decisions by fuzzy templates. In: EUFIT, pp. 1470–1474 (1995)
17. Singla, A., Yuan, L., Ebrahimi, T.: Food/non-food image classification and food categorization using pre-trained googlenet model. In: Proceedings of the 2nd International Workshop on MADiMa, pp. 3–11 (2016)
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 (2016)
19. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv Preprint (2016)