# Food Recognition for Dietary Assessment Using Deep Convolutional Neural Networks

Stergios Christodoulidis[1,2(✉)], Marios Anthimopoulos[1,3], and Stavroula Mougiakakou[1,4]

[1] ARTORG Center for Biomedical Engineering Research, University of Bern, Bern, Switzerland
{stergios.christodoulidis,marios.anthimopoulos, stavroula.mougiakakou}@artorg.unibe.ch
[2] Graduate School of Cellular and Biomedical Sciences, University of Bern, Bern, Switzerland
[3] Department of Emergency Medicine, Bern University Hospital, Bern, Switzerland
[4] Department of Endocrinology, Diabetes and Clinical Nutrition, Bern University Hospital, Bern, Switzerland

**Abstract.** Diet management is a key factor for the prevention and treatment of diet-related chronic diseases. Computer vision systems aim to provide automated food intake assessment using meal images. We propose a method for the recognition of already segmented food items in meal images. The method uses a 6-layer deep convolutional neural network to classify food image patches. For each food item, overlapping patches are extracted and classified and the class with the majority of votes is assigned to it. Experiments on a manually annotated dataset with 573 food items justified the choice of the involved components and proved the effectiveness of the proposed system yielding an overall accuracy of 84.9%.

**Keywords:** Food recognition · Convolutional neural networks · Dietary management · Machine learning

## 1 Introduction

Diet-related chronic diseases like obesity and diabetes have become a major health concern over the last decades. Diet management is a key factor for the prevention and treatment of such diseases, however traditional methods often fail due to the inability of patients to assess accurately their food intake. This situation raises an urgent need for novel tools that will provide automatic, personalized and accurate diet assessment. Recently, the widespread use of smartphones with enhanced capabilities together with the advances in computer vision, enabled the development of novel systems for dietary management on mobile phones. Such a system takes as input one or more images of a meal and either classifies them as a whole or segments the food items and recognizes them separately. Portion estimation is also provided by some systems based on the 3D reconstruction of food. Finally, the meal's nutritional content is estimated using

© Springer International Publishing Switzerland 2015
V. Murino et al. (Eds.): ICIAP 2015 Workshops, LNCS 9281, pp. 458–465, 2015.
DOI: 10.1007/978-3-319-23222-5_56

nutritional databases and returned to the user. Here, we focus on food recognition which constitutes the common denominator in this new generation of systems. To this end, various approaches have been proposed derived from the particularly active fields of image classification and object recognition. The problem is usually divided into two tasks: description and classification.

Some systems employed handcrafted global descriptors, capturing mainly color and texture information: quantized color histograms [1, 2], first-order color statistics [3, 4, 5], Gabor filtering [6], [7] and local binary patterns (LBP) [2] have been used among others. In order to achieve a description adapted to the problem, visual codebooks have been utilized, created by clustering local descriptors. The most popular choices for local descriptors are: the classic SIFT [1] and its color variants [9], [10] as well as the histogram of oriented gradients (HoG) [11, 12, 13]. Other kinds of local descriptors include filter banks like the maximum response filters [8], [14] or even raw values of neighboring pixels [15]. Visual codebooks are often created within bag of features (BoF) approaches where image patches are described and assigned to the closest visual word from the codebook, while the resulting histogram constitutes the global descriptor [1], [9], [10], [16]. When filter banks are used for the local description the term texton analysis is used instead [8], [14], [15]. Other approaches attempted to reduce the quantization error introduced by the hard assignment of each patch to a single visual word. Sparse coding was used in [6] which represents patches as sparse linear combinations of visual words. On the other hand, the locality-constrained linear coding (LLC) used in [3], [12] enforces locality instead of sparsity producing smaller coefficients for distant visual words. Finally, the Fisher vector (FV) approach used in [11], [13], [17] fits a Gaussian mixture model (GMM) to the local feature space instead of clustering, and then characterize a patch by its deviation from the GMM distribution. For the classification, the support vector machines (SVM) have been the most popular choice. Gaussian kernels were used in many systems [2], [5] whereas for histogram based features the chi-squared kernel is reported to be the best choice [8], [15]. For highly dimensional features spaces even linear kernels often perform satisfactorily [13]. Finally, multiple kernel learning has also been used for the fusion of different types of features [7], [10].

Recently, an approach based on deep convolutional neural networks (CNN) [18] gained attention by winning the ImageNet Large-Scale Visual Recognition Challenge and outperforming by far the competition. The eight-layer network of [18] was used in [11] for the classification of Japanese food images in 100 classes. However, due to the huge size of the network and the limited amount of images (14,461), the results were not adequate so a FV representation on HoG and RGB values was also employed to provide complementary description. In [20], a four-layer CNN was used for food recognition. A dataset with 170,000 images belonging to 10 classes was created and images were downscaled to 80×80 and then randomly cropped to 64×64 before fed to the CNN.

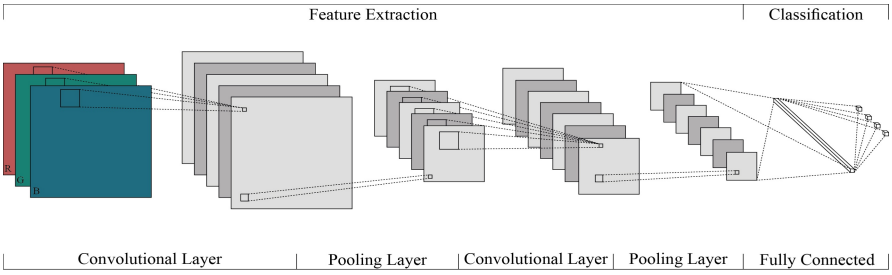


**Fig. 1.** Typical architecture of a convolutional neural network

In this study, we propose a system for the recognition of already segmented food items in meal images using a deep CNN, trained on fixed-size local patches. Our approach exploits the outstanding descriptive ability of a CNN, while the patch-wise model allows the generation of sufficient training samples, provides additional spatial flexibility for the recognition and ignores background pixels.

## 2 Methods

Before describing the architecture and the different components of the proposed system, we provide a brief introduction to the deep CNNs.

### 2.1 Convolutional Neural Networks

CNNs are multi-layered artificial neural networks which incorporate both unsupervised feature extraction and classification. A CNN consists of a series of convolutional and pooling layers that perform feature extraction followed by one or more fully connected layers for the classification. Convolutional layers are characterized by sparse connectivity and weight sharing. The inputs of a unit in a convolutional layer come from just a small rectangular subset of units of the previous layer. In addition, the nodes of a convolutional layer are grouped in feature maps sharing the same weights. The inputs of each feature map are tiled in such a way that correspond to overlapping regions of the previous layer making the aforementioned procedure equivalent to convolution while the shared weights within each map correspond to the kernels . The output of convolution passes through an activation function that produces nonlinearities in an element-wise fashion. A pooling layer follows which subsamples the previous layer by aggregating small rectangular subsets of values. Max or mean pooling is applied replacing the input values with the maximum or the mean value, respectively. A number of fully connected layers follow with the last one having a number of units equal to the number of classes. This part of the network performs the supervised classification and takes as input the values of the last pooling layer which constitute the feature set. For training the CNN a gradient descent method is applied using back propagation. A schematic representation of a CNN with two pairs of convolutional-pooling layers and two fully connected layers is depicted in Fig. 1.

## 2.2 System Description

The proposed system recognizes already segmented food items using an ensemble learning model. For the classification of a food item, a set of overlapping square patches is extracted from the corresponding area on the image and each of them is classified by a CNN into one of the considered food classes. The class with the majority of votes coming from the local classifications is finally assigned to the food item. Our approach is comprised by three main stages: preprocessing, network training and food recognition. An overview of the system is depicted in Fig. 2.

**Preprocessing.** This stage aims at preparing the data for the CNN training procedure. First, non-overlapping patches of size 32×32 are extracted from the inside of each food item in the dataset. In order to increase the amount of training data and prevent over-fitting we artificially augment the training patch dataset by using label-preserving transformations such as flip and rotation as well as the combinations of the two. In total, 16 transformations are used. Then, we calculate the mean over the training image patches and subtract it from all the patches of the dataset so the CNN takes as input mean centered RGB pixel values.

**Network Training.** Using the created patch dataset we train a deep CNN with a six layer architecture. The network has four convolutional layers with 5×5 kernels; the first three layers have 32 kernels while the last has 64, producing equal number of feature maps. All the activation functions are set to the rectified linear unit (ReLU) since it has been reported to minimize the classification error of the network faster than other activation functions such as *tanh* [18]. Each convolutional layer is followed by a

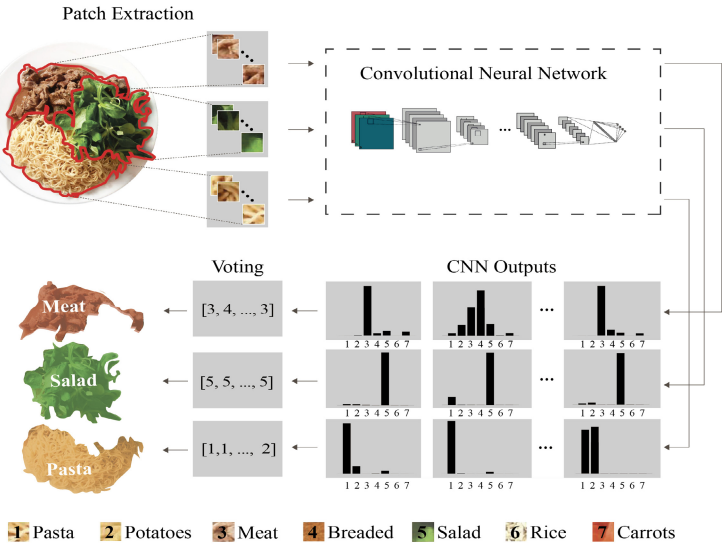


**Fig. 2.** The proposed system overview.

pooling layer with 3×3 pooling regions and stride equal to two; the first one outputs the maximum value out of each pooling region while the following three use the average. The last two layers of the network are fully connected with 128 and 7 units, respectively. On these layers, random dropout of units was used to prevent overfitting [21]. The output of each hidden neuron was set to zero with a probability $p$ forcing the network to learn more robust features for the description of the input regardless of the inactive neurons. Here, the dropout probability $p$ is set to 0.5. The softmax function is used so as to normalize the outputs of the last layer so each output is between zero and one and they all sum up to one. This way, the output values represent a categorical probability distribution so a cross-entropy loss function is used to calculate the error used by gradient descent training. Finally, as far as the weight learning is concerned, a schema with a decay of the learning rate along with a momentum coefficient was used. The base learning rate is set to 0.001 with an exponential decay policy and the momentum is set to 0.9.

**Food Recognition.** For the recognition of the food items a voting scheme is used. For each food item to be classified, images patches are extracted preprocessed and fed to the CNN. The most frequent class occurring from the classification of the patches is then assigned to the food item.

## 3 Experimental Setup and Results

### 3.1 Experimental Setup

For training and testing the proposed system we used a dataset of 246 images of different meals served in the restaurants of Bern University hospital, "Inselspital". The images contain in total 573 food items, belonging to seven broad food classes, namely pasta, potatoes, meat, breaded food, rice, green salad and carrots. For each image an annotation map has been manually created containing the area and the class label of the existing food items. The evaluation procedure for the classification of both patches and food items is based on a 5-fold cross-validation scheme which is applied on a food item level in order to avoid biased results. For each fold, we used the ground truth maps to extract a number of 32×32 patches leading to a set of nearly 160,000 training patches per fold which proved to be sufficient for training the CNN. The performance in the experiments is assessed in terms of average F-score over the different classes in a patch ($pF_{avg}$) or food item level ($F_{avg}$). The total accuracy of the food item classification is also considered. The experiments were conducted in the deep learning framework Caffe [22] using a single GPU (GeForce GTX 760, 2GB Memory, 1152 Cores).

### 3.2 Results

The configuration of the CNN was initially based on the cifar-10 solution[1]. However, in order to find the most suitable configuration for the proposed system, a number of

[1] https://code.google.com/p/cuda-convnet/wiki/Methodology

experiments were conducted on the involved components and their parameters. Table 1 presents the results for the different configurations that were tested. The optimal number of convolutional-pooling layers was four. The use of the dropout technique for the penultimate layer further improved the results. However, the use of a local response normalization (LRN) after the activation functions did not present a clear improvement.

**Table 1.** Results for the different architectures that were investigated. For all the convolutional layers 5x5 kernels was used and for all the pooling layers 3x3 pooling regions. Notation: cp – convolutional-pooling layers, fc – fully connected layers, $pF_{avg}$- the f-score on a patch level.

| CNN architecture | $pF_{avg}$(%) |
|---|---|
| 32cp – 32cp – 128fc – 7fc | 66.5 |
| 32cp – 32cp – 64cp – 128fc – 7fc | 68.7 |
| 32cp – 32cp – 32cp – 64cp – 128fc – 7fc | 69.5 |
| 32cp – 32cp – 32cp – 64cp – 64cp – 128fc – 7fc | 67.1 |
| 32cp – 32cp – 32cp – 64cp – 128fc – 7fc + LRN | 70.4 |
| **32cp – 32cp – 32cp – 64cp – 128fc – 7fc + Dropout** | **71.79** |
| 32cp – 32cp – 32cp – 64cp – 128fc – 7fc + LRN + Dropout | 71.28 |

**Table 2.** Results of the proposed method for different voting schemes and variants compared to a method from the literature

| Classification Method | Accuracy | $F_{avg}$ | Time (sec/item) |
|---|---|---|---|
| Patch-wise CNN + Weighted voting + step=16 | 84.6 | 82.8 | 0.28 |
| Patch-wise CNN + Max voting + step=32 | 83.5 | 81.4 | 0.11 |
| **Patch-wise CNN + Max voting + step=16** | **84.9** | **82.7** | **0.28** |
| Patch-wise CNN + Max voting + step=8 | 84.7 | 82.5 | 0.92 |
| Learned histogram + Multi-scale LBP + SVM | 82.2 | 79.7 | 0.1 |

Fig. 3 presents the 32 convolutional kernels from the first layer of the proposed network. It can be observed that the kernels capture mainly color information which is the primal feature for the discrimination among foods. After configuring the CNN architecture for the classification of patches, we conducted an investigation regarding the best use of this fixed-scale classifier for the recognition of food items. Two are the main involved elements; the voting scheme and the density of the classification. For the voting, we tested two techniques: (i) voting only for the best candidate class (Max

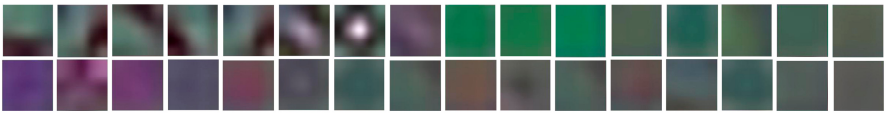

**Fig. 3.** The kernels from the first layer of the proposed CNN

voting) and (ii) voting for all the classes using the CNN output, after the softmax normalization, as weights (weighted voting). For the density of the classified patches on each food item we used several step values resulting in different overlaps. Table 2 shows the corresponding results. As it can be seen, the max voting scheme presented slightly better performance while a maximum overlap of 50% (step = 16) among the extracted patches was proved to be optimal. Table 2 also provides a comparison with a method from the state of the art in the same dataset. The method is based on [2] and uses adapted color histograms and multi-scale LBP features fed to an SVM with a Gaussian kernel. The proposed recognition system scored nearly 3% more in both metrics showing the potential of CNN in the food recognition problem. The average processing time per image for the selected configuration was 0.28 seconds which is more than most conventional methods but still acceptable

## 4 Conclusions

We proposed a method for the recognition of already segmented food items using a CNN. The classification is applied in a patch-wise manner and a voting technique was used to determine the class of each food item. The patch-wise model together with the data augmentation trick allowed us to extract a sufficient amount of samples to train a 6-layer CNN. The experimental results proved the effectiveness of the system that achieved an overall accuracy of 84.9%. The presented results are preliminary; future work should include a more thorough investigation on the optimal architecture as well as the training parameters of the network. Moreover, the use of alternative classifiers combined with the CNN features could further enhance the performance.

**Acknowledgments.** This work was funded in part by the Bern University Hospital "Inselspital", the European Union Seventh Framework Programme (FP7-PEOPLE-2011-IAPsP) under grant agreement n° 286408 [www.gocarb.eu] and the Swiss National Science Foundation n° 156511 [p3.snf.ch/project-156511].

## References

1. Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R.,Yang, J.: PFID: Pittsburgh fast-food image dataset. In: 16th IEEE International Conference on Image Processing (2009)
2. Anthimopoulos, M., Dehais, J., Diem, P., Mougiakakou, S.: Segmentation and recognition of multi-food meal images for carbohydrate counting. In: IEEE BIBE (2013)
3. Aizawa, K., Maruyama, Y., He, L., Morikawa, C.: Food Balance Estimation by Using Personal Dietary Tendencies in a Multimedia Food Log. IEEE Transactions on Multimedia **15**(8), 2176–2185 (2013)
4. Fengqing, Z., Bosch, M., Khanna, N., Boushey, C.J., Delp, E.J.: Multiple Hypotheses Image Segmentation and Classification With Application to Dietary Assessment. IEEE Journal of Biomedical and Health Informatics **19**(1), 377–388 (2015)

5. Oliveira, L., Costa, V., Neves, G., Oliveira, T., Jorge, E., Lizarraga, M.: A mobile, lightweight, poll-based food identification system. Pattern Recognition **47**, 1941–1952 (2014)
6. Chen, M.Y., Yang, Y.H., Ho, C.J., Wang, S.H., Liu, S.M., Chang, E., Yeh, C.H., Ouhyoung, M.: Automatic chinese food identification and quantity estimation. In: SIGGRAPH Asia 2012 (2012)
7. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In: IEEE International Conference on Multimedia and Expo (2012)
8. Puri, M., Zhu, Z., Yu, Q., Divakaran, A., Sawhney, H.: Recognition and volume estimation of food intake using a mobile device. In: IEEE WACV, pp. 1–8 (2009)
9. Anthimopoulos, M.M., Gianola, L., Scarnato, L., Diem, P., Mougiakakou, S.G.: A Food Recognition System for Diabetic Patients Based on an Optimized Bag-of-Features Model. IEEE Journal of Biomedical and Health Informatics **18**(4), 1261–1271 (2014)
10. Bettadapura, V., Thomaz, E., Parnami, A., Abowd, G.D., Essa, I.A.: Leveraging context to support automated food recognition in restaurants. In: WACV 2015, pp. 580–587 (2015)
11. Kawano, Y., Yanai, K.: Food image recognition with deep convolutional features. In: ACM UbiComp Workshop on Cooking and Eating Activities (CEA) (2014)
12. Beijbom, O., Joshi, N., Morris, D., Saponas, S., Khullar, S.: Menu-match: restaurant-specific food logging from images. In: 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 844–851 (2015)
13. Kawano, Y., Yanai, K.: FoodCam: A real-time food recognition system on a smartphone. Multimedia Tools and Applications (2014)
14. Farinella, G.M., Moltisanti, M., Battiato, S.: Classifying food images represented as bag of textons. In: IEEE International Conference on Image Processing (ICIP), pp. 5212–5216 (2014)
15. Yang, S., Chen, M., Pomerleau, D., Sukthankar, R.: Food recognition using statistics of pairwise local features. In: CVPR 2010 (2010)
16. Nguyen, D.T., Zong, Z., Ogunbona, P., Probst, Y.C., Li, W.: Food image classification using local appearance and global structural information. Neurocomputing **140**, 242–251 (2014)
17. 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, Part VI. LNCS, vol. 8694, pp. 446–461. Springer, Heidelberg (2014)
18. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NIPS 2012 (2012)
19. 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 (2014)
20. Kagaya, H., Aizawa, K., Ogawa, K.: Food Detection and Recognition Using Convolutional Neural Network. ACM Multimedia, 1085–1088 (2014)
21. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. http://arxiv.org/abs/1207.0580
22. Yangqing, J.: Caffe: An open source convolutional architecture for fast feature embedding (2013). http://caffe.berkeleyvision.org