FoodNet: Recognizing Foods Using Ensemble of Deep Networks

Paritosh Pandey*, Akella Deepthi*, Bappaditya Mandal and N. B. Puhan

***Abstract*—In this paper we propose a protocol for an automatic food recognition system which identifies the contents of the meal from the images of the food. We developed a multi-layered convolutional neural network (CNN) pipeline that takes advantages of the features from other deep networks and improves the efficiency. Numerous traditional handcrafted features and methods are explored, among which CNNs are chosen as the best performing features. Networks are trained and fine-tuned using preprocessed images and the filter outputs are fused to achieve higher accuracy. Experimental results on the largest real-world food recognition database ETH Food-101 and newly contributed Indian food image database demonstrate the effectiveness of the proposed methodology as compared to many other benchmark deep learned CNN frameworks.**

***Index Terms*—Deep CNN, Food Recognition, Ensemble of Networks, Indian Food Database.**

I. Introduction and Current Approaches

THERE has been a clear cut increase in the health consciousness of the global urban community in the previous few decades. Given the rising number of cases of health problems attributed to obesity and diabetes reported every year, people (including elderly, blind or semi-blind or dementia patients) are forced to record, recognize and estimate calories in their meals. Also, in the emerging social networking photo sharing, food constitutes a major portion of these images. Consequently, there is a rise in the market potential for such fitness apps products which cater to the demand of logging and tracking the amount of calories consumed, such as [1], [2]. Food items generally tend to show intra-class variation depending upon the method of preparation, which in turn is highly dependent on the local flavors as well as the ingredients used. This causes large variations in terms of shape, size, texture, and color. Food items also do not exhibit any distinctive spatial layout. Variable lighting conditions and the point of view also lead to intra-class variations, thus making the classification problem even more difficult [3]. Hence food recognition is a challenging task, one that needs addressing.

In the existing literature, numerous methodologies assume that the texture, color and shape of food items are well defined [4], [5], [6]. This may not be true because of the local variations in the method of food preparation, as well as the ingredients used. Feature descriptors like histogram of gradient (HOG), color correlogram, bag of scale-invariant feature transform (SIFT), local binary pattern (LBP), spatial pyramidal pooling, speeded up robust features (SURF), etc, have been applied with some success on small laboratory generated datasets [6]. Hoashi *et al.* in [7], and Joutou *et al.* in [8] propose multiple kernel learning methods to combine various feature descriptors. The features extracted have generally been used to train an SVM [9], with a combination of these features being used to boost the accuracy.

A rough estimation of the region in which targeted food item is present would help to raise the accuracy for cases with non-uniform background, presence of other objects and multiple food items [10]. Two such approaches use standard segmentation and object detection methods [11] or asking the user to input a bounding box providing this information [12]. Kawano *et al.* [12], [13] proposed a semi-automated approach for bounding box formation around the image and developed a real-time recognition system. It is tedious, unmanageable and does not cater to the need of full automation. Automatic recognition of dishes would not only help users effortlessly organize their extensive photo collections but would also help online photo repositories make their content more accessible. Lukas *et al.* in [14] have used a random forest to find discriminative region in an image and have shown to under perform convolutional neural network (CNN) feature based method [15].

In order to improve the accuracy, Bettadapura *et al.* in [16] used geotagging to identify the restaurant and search for matching food item in its menu. Matsuda *et al.* in [17] employed co-occurrence statistics to classify multiple food items in an image by eliminating improbable combinations. There has been certain progress in using ingredient level features [18], [19], [20] to identify the food item. A variant of this method is the usage of pairwise statistics of local features [21]. In the recent years CNN based classification has shown promise producing excellent results even on large and diverse databases with non-uniform background. Notably, deep CNN based transferred learning using fine-tuned networks is used in [22], [23] and cascaded CNN networks are used in [24]. In this work, we extend the CNN based approaches towards combining multiple networks and extract robust food discriminative features. We have prepared a new Indian food image database for this purpose, the largest to our knowledge and experimented on two large databases, which demonstrates the effectiveness of the proposed framework. We will make all the developed models and Indian food database available online to public. Section II describes our proposed methodology and Section III provides the experimental results before drawing conclusions in Section IV.

P. Pandey, A. Deepthi and N. B. Puhan are with the School of Electrical Science, Indian Institute of Technology (IIT), Bhubaneswar, Odisha 751013, India. E-mail: {pp20, da10, nbpuhan}@iitbbs.ac.in

B. Mandal is with the Institute for Infocomm Research (I^2R), A*STAR, Singapore 138632. Email: bmandal@i2r.a-star.edu.sg

* Represents equal contribution from the authors.

II. Proposed Method

Our proposed framework is based on recent emerging very large deep CNNs. We have selected CNNs because their ability to learn operations on visual data is extremely good and they have been employed to obtain higher and higher accuracies on challenges involving large scale image data [25]. We have performed extensive experiments using different handcrafted features (such as HOG, color correlogram, bag of SIFT, LBP, Spatial Pyramidal Pooling, SURF, etc) and CNN feature descriptors. Experimental results show that CNNs outperform all the other methods by a huge margin, similar to those reported in [6]. In the following subsections, we describe our ensemble of deep CNN network architecture in details.

A. Proposed Ensemble Network Architecture

We choose AlexNet architecture by Krizhevsky *et al.* [15] as our baseline because it offers the best solution in terms of significantly lesser computational time as compared to any other state-of-the-art CNN classifier. GoogLeNet architecture by Szegedy *et al.* [26] uses the sparsity of the data to create dense representations that give information about the image with finer details. It develops a network that would be deep enough, as it increases accuracy and yet have significantly less parameters to train. This network is an approximation of the sparse structure of a convolution network by dense components. The building blocks called Inception modules, is basically a concatenation of filter banks with a mask size of 1×1, 3×3 and 5×5. If the network is too deep, the inception modules lead to an unprecedented rise in the cost of computation. Therefore, 1×1 convolutions are used to embed the data output from the previous layers.

ResNet architecture by He *et al.* [27] addresses the problem of degradation of learning in networks that are very deep. In essence a ResNet is learning on residual functions of the input rather than unreferenced functions. The idea is to reformulate the learning problem into one that is easier for the network to learn. This is to say, the original problem of learning a function $H(x)$ gets transformed into learning non-linearly by various layers fitting the functional form $H(x) = \Gamma(x) + x$, which is easier to learn, where the layers have already learned $\Gamma(x)$ and the original input is x. These CNN networks are revolutionary in the sense that they were at the top of the leader board of ImageNet classification at one or other time [25], with ResNet being the network with maximum accuracy at the time of writing this paper. The main idea behind employing these networks is to compare the increment in accuracies with the depth of the network and the number of parameters involved in training. Our idea is to create an ensemble of these classifiers using another CNN on the lines of a Siamese network [28] and other deep network combinations [29].

In a Siamese network [28], two or more identical subnetworks are contained within a larger network. These subnetworks have the same configuration and weights. It has been used to find comparisons or relationships between the two input objects or patches. In our architecture, we use this idea to develop a three layered structure to combine the feature outputs of three different subsections (or subnetworks) as shown in Fig. 1. We hypothesize that these subnetworks with proper fine-tuning would individually contribute to extract better discriminative features from the food images. However, the parameters along with the subnetwork architectures are different and the task is not that of comparison (as in case of Siamese network [28]) but pursue classification of food images. Our proposition is that the features once added with appropriate weights would give better classification accuracies.

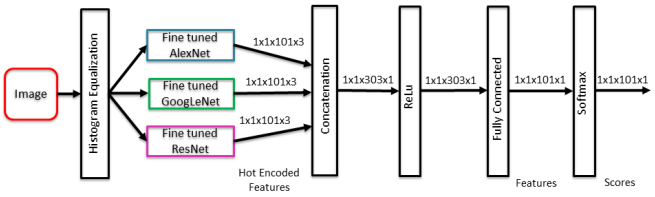

Fig. 1. Our proposed CNN based ensemble network architecture.

Let $I(w, h, c)$ represents a pre-processed input image of size $w \times h$ pixels to each of the three fine-tuned networks and c is the number of channels of the image. Color images are used in our case. We denote $C(m, n, q)$ as the convolutional layer, where m and n are the sides length of the receptive field and q is the number of filter banks. Pooling layer is denoted by $P(s, r)$, where r is the side length of the pooling receptive field and s is the number of strides used in our CNN model. In our ensemble net we did not use pooling. But in our fine-tuned networks pooling is employed with variable parameters. GoogLeNet for example uses overlapping pooling in the inception module. All convolution layers are followed by ReLU layers (see the text in Sec II-B) considered as an in-built activation. L represents the local response normalization layer. Fully connected layer is denoted by $F(e)$, where e is the number of neurons. Hence, the AlexNet CNN model after fine-tuning is represented as:

$$\begin{aligned}\Phi_A \equiv\ & I(227, 227, 3) \longrightarrow C(11, 4, 96) \longrightarrow L \longrightarrow P(2, 3) \longrightarrow C(5, 1, 256) \\ & \longrightarrow L \longrightarrow P(2, 3) \longrightarrow C(3, 1, 384) \longrightarrow C(3, 1, 384) \longrightarrow C(3, 1, 256) \\ & \longrightarrow P(2, 3) \longrightarrow F(4096) \longrightarrow F(4096) \longrightarrow F(e).\end{aligned} \tag{1}$$

AlexNet is trained in a parallel fashion, referred as a depth of 2. Details of the architecture can be found in [15]. For GoogLeNet we need to define the inception module as: $D(c1, cr3, c3, cr5, c5, crM)$, where $c1$, $c3$ and $c5$ represent no of filter of size 1×1, 3×3 and 5×5, respectively. $cr3$ and $cr5$ represent no of 1×1 filters used in the reduction layer prior to 3×3 and 5×5 filters, and crM represents the number of 1×1 filters used as reduction after the built in max pool layer. Hence GoogLeNet is fine-tuned as:

$$\begin{aligned}\Phi_G \equiv\ & I(224, 224, 3) \longrightarrow C(7, 2, 64) \longrightarrow P(2, 3) \longrightarrow L \longrightarrow C(1, 1, 64) \\ & \longrightarrow (3, 1, 1-92) \longrightarrow L \longrightarrow P(2, 3) \longrightarrow D(64, 96, 128, 16, 32, 32) \\ & \longrightarrow D(128, 128, 192, 32, 96, 64) \longrightarrow P(2, 3) \longrightarrow \\ & D(192, 96, 208, 16, 48, 64) \longrightarrow D(160, 112, 224, 24, 64, 64) \longrightarrow \\ & D(128, 128, 256, 24, 64, 64) \longrightarrow D(112, 144, 288, 32, 64, 64) \longrightarrow \\ & D(256, 160, 320, 32, 128, 128) \longrightarrow P(2, 3) \longrightarrow D(256, 160, 320, 32, \\ & 128, 128) \longrightarrow D(384, 192, 384, 48, 128, 128) \longrightarrow P^*(1, 7) \longrightarrow F(e),\end{aligned} \tag{2}$$

P^* refers to average pooling rather than max pooling used everywhere else. For fine-tuned ResNet, each repetitive residual

unit is presented inside as R and it is defined as:

$$\begin{aligned}\Phi_R \equiv I(224,224,3) &\longrightarrow C(7,2,64) \longrightarrow P(2,3) \longrightarrow 3 \times R(C(1,1,64) \\ &\longrightarrow C(3,1,64) \longrightarrow C(1,1,256)) \longrightarrow R(C(1,2,128) \longrightarrow C(3,2,128) \\ &\longrightarrow C(1,2,512)) \longrightarrow 3 \times R(C(1,1,128) \longrightarrow C(3,1,128) \\ &\longrightarrow C(1,1,512)) \longrightarrow R(C(1,2,256) \longrightarrow C(3,2,256) \longrightarrow \\ &C(1,2,1024)) \longrightarrow 5 \times R(C(1,1,256) \longrightarrow C(3,1,256) \longrightarrow \\ &C(1,1,1024)) \longrightarrow R(C(1,2,512) \longrightarrow C(3,2,512) \longrightarrow C(1,2,2048)) \\ &\longrightarrow 2 \times R(C(1,1,512) \longrightarrow C(3,1,512) \longrightarrow C(1,1,2048)) \\ &\longrightarrow P^*(1,7) \longrightarrow F(e).\end{aligned} \tag{3}$$

Batch norm is used after every convolution layer in ResNet. The summations at the end of each residual unit are followed by a ReLU unit. For all cases, the length of $F(e)$ depends on the number of categories to classify. In our case, e is the number of classes. Let F_i denote the features from each of the fine-tuned deep CNNs given by (1)-(3), where $i \in \{A, G, R\}$. Let the concatenated features are represented by $\Omega(O, c)$, where O is the output features from all the networks, given by:

$$O = concatenate(w_i F_i) \mid \forall\ i, \tag{4}$$

where w_i is the weight given to features from each of the networks with the constraint, such that $\Sigma_i w_i = 1$.

We define the developed ensemble net as the following:

$$\Phi_E \equiv \Omega(e * \eta, c) \longrightarrow ReLU \longrightarrow F(e) \longrightarrow SoftMax, \tag{5}$$

where η is the number of fine-tuned networks. The $SoftMax$ function or the normalized exponential function is defined as:

$$S(F)_j = \frac{exp^{F_j}}{\sum_{k=1}^{e} exp^{F_k}}, \text{for } j = 1, 2, \ldots, e, \tag{6}$$

where exp is the exponential. The final class prediction $D \in \{1, 2, \ldots, e\}$ is obtained by finding the $maximum$ of the values of $S(F)_j$, given by:

$$D = \arg\max_j (S(F)_j), \text{for } j = 1, 2, \ldots, e. \tag{7}$$

B. Network Details

The ensemble net we designed consists of three layers as shown in Fig. 1. Preprocessed food images are used to fine-tune all the three CNN networks: AlexNet, GoogLeNet and ResNet. Then the first new layer one concatenates the features obtained from the previously networks, passing it out with a rectified linear unit (ReLU) non-linear activation. The outputs are then passed to a fully connected (fc) layer that convolves the outputs to the desired length of the number of classes present. This is followed by a softmax layer which computes the scores obtained by each class for the input image.

The pre-trained models are used to extract features and train a linear kernel support vector machine (SVM). The feature outputs of the fully connected layers and max-pool layers of AlexNet and GoogLeNet are chosen as features for training and testing the classifiers. For feature extraction, the images are resized and normalized as per the requirement of the networks. For AlexNet we used the last fully connected layer to extract features (fc7) and for GoogLeNet we used last max pool layer (cls3_pool). On the ETH Food 101 database, the top-1 accuracy obtained remained in the range of 39.6% for AlexNet to 44.06% for GoogLeNet, with a feature size varying from a minimum of 1000 features per image to 4096 features per image. Feature length of the features extracted out of the last layer is 1000. The feature length out of the penultimate layer of AlexNet gave a feature length of 4096 features, while the ones out of GoogLeNet had a feature length of 1024. All the three networks are fine-tuned using the ETH Food-101 database. The last layer of filters is removed from the network and replaced with an equivalent filter giving an output of the size $1 \times 1 \times 101$, i.e., a single value for 101 channels. These numbers are interpreted as scores for each of the food class in the dataset. Consequently, we see a decrease in the feature size from 1×1000 for each image to 1×101 for each image. AlexNet is trained for a total of 16 epochs.

We choose the MatConvNet [30] implementation of GoogLeNet with maximum depth and maximum number of blocks. The implementation consists of 100 layers and 152 blocks, with 9 Inception modules (very deep!). To train GoogLeNet, the deepest softmax layer is chosen to calculate objective while the other two are removed. The training ran for a total of 20 epochs. ResNet's smallest MatConvNet model with 50 layers and 175 blocks is used. The capacity to use any deeper model is limited by the capacity of our hardware. The batch size is reduced to 32 images for the same reason. ResNet is trained with the data for 20 epochs. The accuracy obtained increased with the depth of the network. The ensemble net is trained with normalized features/outputs of the above three networks. Parametrically weights are decided for each network feature by running the experiments multiple times. A total of 30 epochs are performed. A similar approach is followed while fine-tuning the network for Indian dataset. As the number of images is not very high, jitters are introduced in the network to make sure the network remains robust to changes. Same depth and parameters are used for the networks. The output feature has a length of $1 \times 1 \times 50$ implying a score for each of the 50 classes.

III. Experimental Setup and Results

The experiments are performed on a high end server with 128GB of RAM equipped with a NVDIA Quadro K4200 with 4GB of memory and 1344 CUDA cores. We performed the experiments on MATLAB 14a using the MatConvNet library offered by vlFeat [31]. Caffe's pre-trained network models imported in MatConvNet are used. We perform experiments on two databases: ETH Food-101 Database and and our own newly contributed Indian Food Database.

A. Results on ETH Food-101 Database

ETH Food-101 [14] is the largest real-world food recognition database consisting of 1000 images per food class picked randomly from foodspotting.com, comprising of 101 different classes of food. So there are 101,000 food images in total, sample images can be seen in [14]. The top 101 most popular and consistently named dishes are chosen and randomly sampled 750 training images per class are extracted. Additionally, 250 test images are collected for each class, and are manually cleaned. Purposefully, the training images are not

Fig. 2. Top row: 10 sample Indian food images. Bottom two rows: one of the food samples (1 class) variations (20 images).

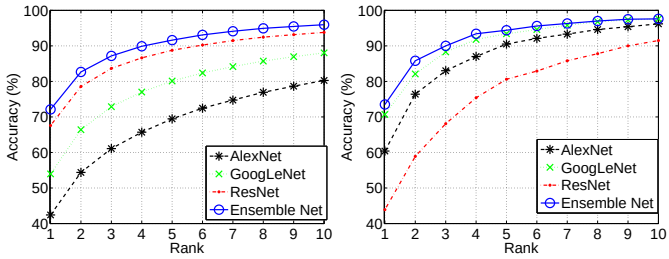

Fig. 3. Rank vs Accuracy plots using various CNN frameworks, (a) for ETH Food 101 Database and (b) for Indian Food Database.

cleaned, and thus contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels to increase the robustness of the data. All images are rescaled to have a maximum side length of 512 pixels. In all our experiments we follow the same training and testing protocols as that in [14], [6].

All the real-world RGB food images are converted to HSV format and histogram equalization are applied on only the intensity channel. The result is then converted back to RGB format. This is done to ensure that the color characteristics of the image does not change because of the operation and alleviate any bias that could have been present in the data due to intensity/illumination variations.

TABLE I
ACCURACY (%) FOR ETH FOOD-101 AND COMPARISON WITH OTHER METHODS AFTER FINE-TUNING.

Network/Features	Top-1	Top-5	Top-10
AlexNet	42.42	69.46	80.26
GoogLeNet	53.96	80.11	88.04
Lukas *et al.* [14]	50.76	-	-
Kawano *et al.* [12]	53.50	81.60	89.70
Martinel *et al.* [6]	55.89	80.25	89.10
ResNet	67.59	88.76	93.79
Ensemble Net	**72.12**	**91.61**	**95.95**

Table I shows the Top-1, Top-5 and Top-10 accuracies using numerous current state-of-the-art methodologies on this database. We have noted only the highest performers, many more results can be found in [6]. It is evident that with fine-tuning the network performance has increased to a large extent. Fig. 3 (a) shows accuracies with the ranks plot up to top 10. From Table I and Fig. 3 (a), it is evident that our proposed ensemble net has outperformed consistently all the current state-of-the-art methodologies on this largest real-world food database.

B. Results on Indian Food Database

One of the contributions of this paper is the setting up of an Indian food database, the first of its kind. It consists of 50 food classes having 100 images each. Some sample images are shown in Fig. 2. The classes are selected keeping in mind the varied nature of Indian cuisine. They differ in terms of color, texture, shape and size as the Indian food lacks any kind of generalized layout. We have ensured a healthy mix of dishes from all parts of the country giving this database a true representative nature. Because of the varied nature of the classes present in the database, it offers the best option to test a protocol and classifier for its robustness and accuracy. We collected images from online sources like foodspotting.com, Google search, as well as our own captured images using hand-held mobile devices. Extreme care was taken to remove any kind of watermarking from the images. Images with textual patterns are cropped, most of the noisy images discarded and a clean dataset is prepared. We also ensured that all the images are of a minimum size. No upper bound on image size has been set. Similar to the ETH Food-101 database protocol, we have randomly selected 80 food images per class for 50 food classes in the training and remaining in the test dataset.

TABLE II
ACCURACY (%) FOR INDIAN FOOD DATABASE AND COMPARISON WITH OTHER METHODS AFTER FINE-TUNING.

Network/Features	Top-1	Top-5	Top-10
AlexNet	60.40	90.50	96.20
GoogLeNet	70.70	93.40	97.60
ResNet	43.90	80.60	91.50
Ensemble Net	**73.50**	**94.40**	**97.60**

Fig. 3 (b) shows accuracies with the ranks plot up to top 10 and Table II shows the Top-1, Top-5 and Top-10 accuracies using some of the current state-of-the-art methodologies on this database. Both these depict that our proposed ensemble of the networks (Ensemble Net) is better at recognizing food images as compared to that of the individual networks. ResNet under performs as compared to GoogLeNet and AlexNet probably because of the lack of sufficient training images to train the network parameters. For overall summary: as is evident from these figures (Fig. 3 (a) and (b)) and tables (Tables I and II) that there is no single second best method that outperforms all others methods in both the databases, however, our proposed approach (Ensemble Net) outperforms all other methods consistently for all different ranks in both the databases.

IV. CONCLUSIONS

Food recognition is a very crucial step for calorie estimation in food images. We have proposed a multi-layered ensemble of networks that take advantages of three deep CNN fine-tined subnetworks. We have shown that these subnetworks with proper fine-tuning would individually contribute to extract better discriminative features from the food images. However, in these subnetworks the parameters are different, the sub-network architectures and tasks are different. Our proposed ensemble architecture outputs robust discriminative features as compared to the individual networks. We have contributed a new Indian Food Database, that would be made available to public for further evaluation and enrichment. We have conducted experiments on the largest real-world food images ETH Food-101 Database and Indian Food Database. The experimental results show that our proposed ensemble net approach outperforms consistently all other current state-of-the-art methodologies for all the ranks in both the databases.

REFERENCES

[1] MealSnap:, "Magical meal logging for iphone," 2017. [Online]. Available: http://mealsnap.com

[2] Eatly:, "Eat smart (snap a photo of your meal and get health ratings)," 2017. [Online]. Available: https://itunes.apple.com/us/app/eatly-eat-smart-snap-photo/id661113749

[3] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. N. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2calories: Towards an automated mobile vision food diary," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1233–1241.

[4] S. Sasano, X. H. Han, and Y. W. Chen, "Food recognition by combined bags of color features and texture features," in 9^{th} *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct 2016, pp. 815–819.

[5] C. Pham and T. N. T. Thanh, "Fresh food recognition using feature fusion," in *International Conference on Advanced Technologies for Communications*, Oct 2014, pp. 298–302.

[6] N. Martinel, C. Piciarelli, and C. Micheloni, "A supervised extreme learning committee for food recognition," *Computer Vision and Image Understanding*, vol. 148, pp. 67–86, 2016.

[7] H. Hajime, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *IEEE International Symposium on Multimedia (ISM)*, Dec 2010.

[8] T. Joutou and K. Yanai, "A food recognition system with multiple kernel learning," in *IEEE International Conference on Image Processing*, Nov 2009.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[10] S. Liu, D. He, and X. Liang, "An improved hybrid model for automatic salient region detection," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 207–210, Apr 2012.

[11] M. Bolaños and P. Radeva, "Simultaneous food localization and recognition," *CoRR*, vol. abs/1604.07953, 2016. [Online]. Available: http://arxiv.org/abs/1604.07953

[12] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun 2013.

[13] K. Yanai and 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)*, June 2015, pp. 1–6.

[14] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014, pp. 446–461.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[16] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa, "Leveraging context to support automated food recognition in restaurants," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 580–587.

[17] M. Yuji and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Proceedings of the* 21^{st} *International Conference on Pattern Recognition*, Nov 2012.

[18] X. Wang, D. Kumar, N. Thorne, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Jul 2015.

[19] J. Chen and C.-w. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *Proceedings of the ACM on Multimedia Conference*, 2016, pp. 32–41.

[20] J. Baxter, "Food recognition using ingredient-level features." [Online]. Available: http://jaybaxter.net/6869_food_project.pdf

[21] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2010.

[22] S. Zhang, H. Yang, and Z.-P. Yin, "Transferred deep convolutional neural network features for extensive facial landmark localization," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 478–482, Apr 2016.

[23] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, Dec 2016.

[24] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015.

[26] C. Szegedy and *et al.*, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] H. Kaiming and *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS)*, 1993, pp. 737–744.

[29] H. Zuo, H. Fan, E. Blasch, and H. Ling, "Combining convolutional and recurrent neural networks for human skin detection," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 289–293, Mar 2017.

[30] VLFEAT, "Vlfeat open source," 2017. [Online]. Available: http://www.vlfeat.org/matconvnet/

[31] ——, "Vlfeat open source," 2017. [Online]. Available: www.vlfeat.org