Technical Report: Exploiting Deep-Features Diversity in Food-11 Classification

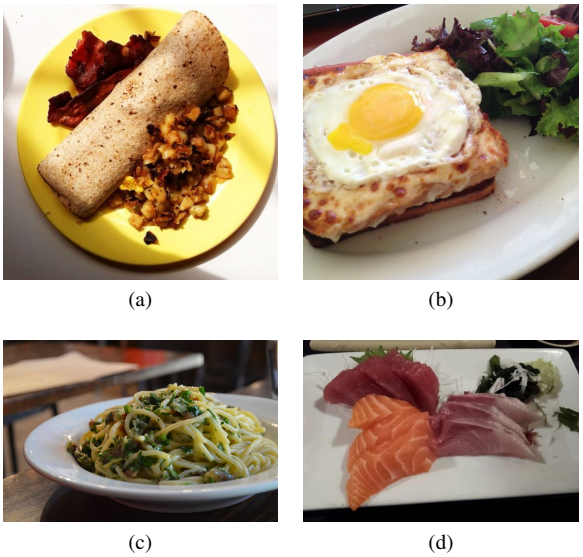
Alan Carvalho Neves*, Caio Cesar Viana da Silva†, Jéssica Soares‡ and Sérgio José de Sousa§
Department of Computer Science
Universidade Federal de Minas Gerais, Brazil
Email: *alan.neves@dcc.ufmg.br, †caiosilva@ufmg.br, ‡jessicasoares@dcc.ufmg.br, §sergiosjs@ufmg.br

I. Introduction

Since the seminal work of *Krizhevsky et. al* [1], many applications in computer vision made use of convolution neural networks. ConvNets tries to model the brain way to understand images, being *state-of-art* in many of this applications.

To perform classification, at each convolutional layer of a ConvNet, filters were created with increasingly complexity, detecting corresponding characteristics of data. As long as the refinement detail increases, deeper a network becomes and more data are needed to the convergence of the weights of the network. This way, ConvNet usually needs of a large amount of data and are computationally expensive to train. However, the results in classification tasks are surprising.

Instead of using ConvNets for classification, some works in literature use them as feature extractor. This is possible due to the nature of deep-features. In this work, we exploit the feature extraction of AlexNet ConvNet [1], pre-trained on ImageNet [2] dataset. We extract the features from layers C1, C5, and FC2 using the Food-11 dataset and use traditional methods of classification, to ensure the expressive power of extracted features.



(a) (b) (c) (d)

Fig. 1: Examples of images from Food-11 dataset: Bread, Egg, Noodles/Pasta and Seafood

II. Experimental Setup

A. Dataset

The dataset of this work is called Food-11 and was released and maintained by *Ecole Polytechnique Fédérale de Lausanne* (EPFL)[1]. This is a dataset containing 16643 food images grouped into 11 major food categories. The 11 categories are Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles/Pasta, Rice, Seafood, Soup, and Vegetable/Fruit. In figure1 we have some samples. The dataset is also divided into three parts: training, validation, and evaluation.

The domain of this data set is challenging, as many kinds of foods are similar - leading to mismatching. Another problem is unbalancing of classes.

B. Implementation details

Our experiments were developed in Python language mainly using three popular libraries: Keras, Theano and scikit-learn.

In detail, Keras [2] is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. Meanwhile, Theano [3] is a library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Last, Scikit-learn [4] is machine learning library, used to run all experiments along this work.

Keras was used to perform feature extraction with pre-trained AlexNet [1] on ImageNet. In this context, Theano was used as backend for Keras and scikit-learn provided many classifiers like SVM, RandomForests and others.

The machine used in experiments has an Intel i7 870 @ 2.93GHz CPU with 16Gb of RAM and a NVIDIA GeForce GTX 1060 GPU.

All implemented code is available to download on our **repository**.

[1]https://mmspg.epfl.ch/food-image-datasets

[2]https://keras.io/

[3]http://deeplearning.net/software/theano/

[4]http://scikit-learn.org/

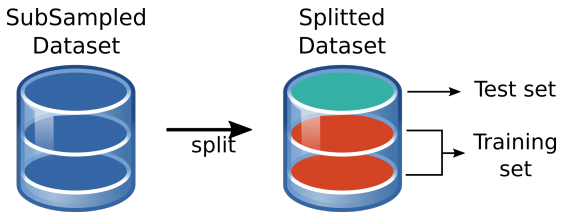


Fig. 2: Division of dataset experiments

III. Results and Discussion

In this section, we present our experiments and the analysis of the results. Some experiments involve varying parameters as well the combination of deep features, in order to evaluate their representational power.

Due to time and resources constraint, we made a subsampling of the dataset. 100 instances of each class were selected in an arbitrary way and formed a balanced new dataset λ.

To perform diversity in our experiments, the data in λ was shuffled and split in two independent sets: two-thirds for *training set* and the remaining to *test set*. A 5-fold cross-validation was performed on the training set to estimate overall accuracy as shown in figure 2. The average accuracy was obtained by averaging the accuracy of each class on testing phase.

Tables containing the accuracy results of the experiments as well heatmaps were built - aiding the analysis process. The heatmap shows how the classifier hits or miss detections in the test phase.

A. Deep features comparison

In this experiment, we perform a comparison between three layers on AlexNet network, in order to test the representational power of them by comparing the values of avarage accuracy (AA) and overall accuracy (OA). As seen in figure 3 and table I, the best classification result was using features of FC2 layer.

On a CNN it is expected that complex characteristics were detected along we walk through the network, due the specialization of filters. As expected, FC2 achieves the best result - these features are complex enough to distinct objects. Thus, for classification tasks, the features extracted by earlier layers would not be the most appropriated. These features would be so simple.

TABLE I: Classification results with different deep representation levels.

Approach	O.A. (%)	A.A. (%)
$C1$	0.376 ± 0.023	0.387 ± 0.171
$C5$	0.643 ± 0.098	0.597 ± 0.131
$FC2$	0.673 ± 0.041	0.638 ± 0.157

B. Deep features combination

The combination of deep features involves testing the descriptive power of combined features. In early fusion, features are merged and sent to the classifier. In late fusion, three classifiers were trained, each one with one deep feature. The result is the majority vote of the classifiers.

The results on table II and figure 4 shows that combination of features in early and late approaches were not sufficient to outperform the result of the last fully connected layer (FC2) - in general. The high average accuracy of late fusion can suggest mismatch on some classes, leading in general poor results.

TABLE II: Classification results by using early fusion, late fusion, and FC2.

Approach	O.A. (%)	A.A. (%)
Early fusion	0.482 ± 0.049	0.481 ± 0.143
Late fusion	0.553 ± 0.134	0.684 ± 0.152
FC2	0.673 ± 0.041	0.638 ± 0.157

C. Diversity evaluation

In the last set of experiments, the goal is to test diversity on FC2 layer. We compare different classifiers against all features obtained in the FC2 layer, as subset features in approaches like bagging and majority voting.

The bagging approach achieved the best results of all experiments executed. The bagged classifier often has significantly greater accuracy than a single classifier, as is well known to be robust to noise and overfitting. The increased accuracy occurs due a reduction of variance of individual classifiers. As seen in figure 5 and table III.

TABLE III: Classification results by using ensemble methods in comparison with original deep features.

Approach	O.A. (%)	A.A. (%)
SVM-RBF ($FC2$)	0.673 ± 0.041	0.638 ± 0.157
Random Forest	0.564 ± 0.073	0.510 ± 0.260
Majority Voting	0.623 ± 0.083	0.618 ± 0.164
Bagging	0.821 ± 0.009	0.603 ± 0.185

IV. Conclusions

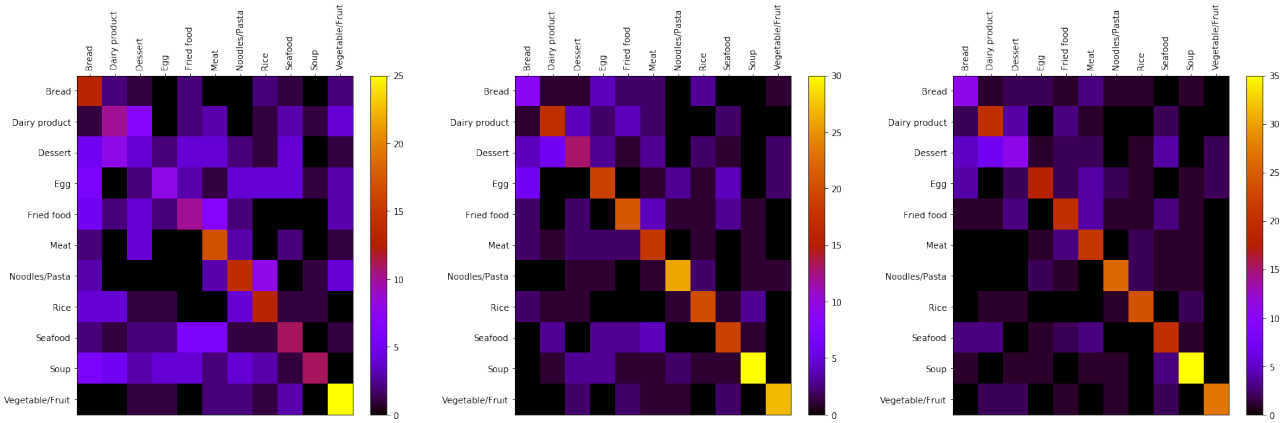
In this work we could learn how the domain could affect classification results. Food domain is hard to classify, even for humans. This happens due to similarity between food made with same ingredients which belongs to different classes.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

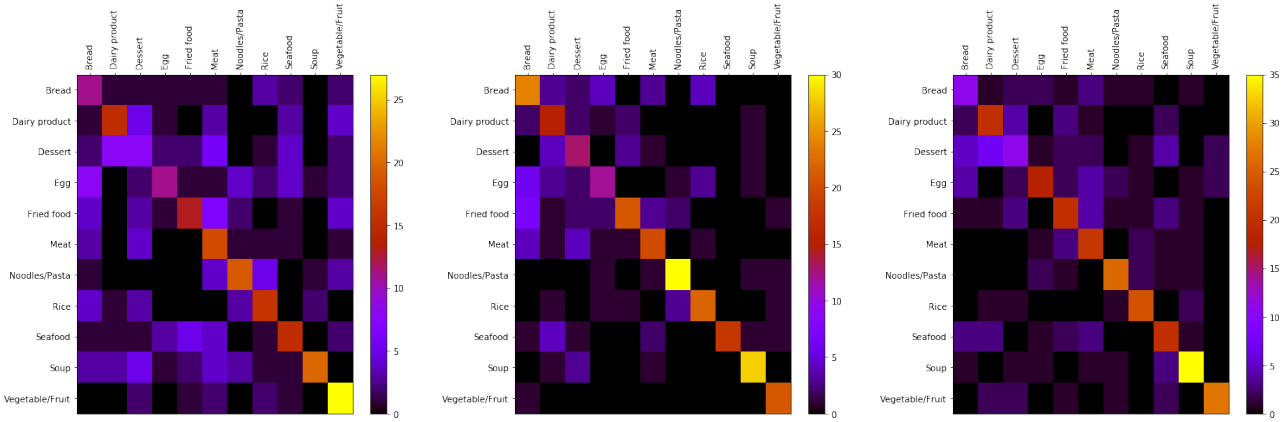
[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[3] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.



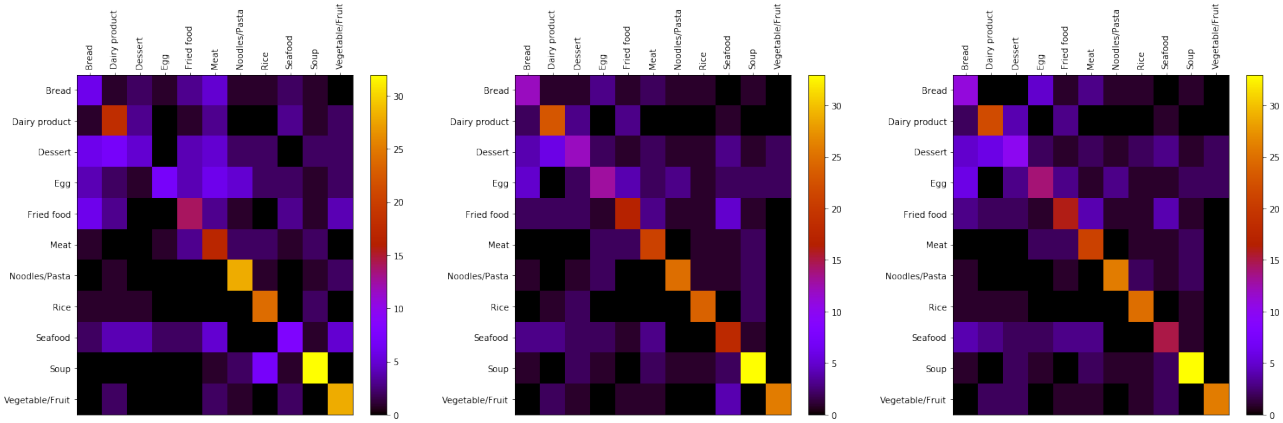
(a) $C1$ (b) $C5$ (c) $FC2$

Fig. 3: Classification results for each class by using SVM-RBF with features $C1$, $C5$, and $FC2$, respectively.



(a) Early fusion (b) Late fusion (c) FC2

Fig. 4: Classification results for each class by using early fusion, late fusion, and $FC2$.



(a) Random Forest (b) Majority Voting (c) Bagging

Fig. 5: Classification results for each class by using ensemble approaches.