Training a Mentee Network by Transferring Knowledge from a Mentor Network

Elnaz Jahani Heravi(✉), Hamed Habibi Aghdam, and Domenec Puig

Department of Computer Engineering and Mathematics,
University Rovira i Virgili, Tarragona, Spain
{elnaz.jahani,hamed.habibi,domenec.puig}@urv.cat

Abstract. Automatic classification of foods is a challenging problem. Results on ImageNet dataset shows that ConvNets are very powerful in modeling natural objects. Nonetheless, it is not trivial to train a ConvNet from scratch for classification of foods. This is due to the fact that ConvNets require large datasets and to our knowledge there is not a large public dataset of foods for this purpose. An alternative solution is to transfer knowledge from already trained ConvNets. In this work, we study how transferable are state-of-art ConvNets to classification of foods. We also propose a method for transferring knowledge from a bigger ConvNet to a smaller ConvNet without decreasing the accuracy. Our experiments on UECFood256 dataset show that state-of-art networks produce comparable results if we start transferring knowledge from an appropriate layer. In addition, we show that our method is able to effectively transfer knowledge to a smaller ConvNet using unlabeled samples.

Keywords: Food classification · Convolutional neural network · Deep learning · Transfer learning

1 Introduction

Obesity is known as a disease in developed countries and it can be controlled by monitoring the food intake. However, accurate calculation of calorie intake is not trivial and patients tend to calculate it quickly and conveniently. Automatic estimation of calorie intake can be done using the image of a food. To this end, first the system recognizes foods in the classification stage and, then, it estimates calorie based on the category of food.

Early attempts on food recognition focused on the traditional approached which extracts features using hand-crafted methods and then applies a classifier

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-49409-8_42) contains supplementary material, which is available to authorized users.

© Springer International Publishing Switzerland 2016
G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 500–507, 2016.
DOI: 10.1007/978-3-319-49409-8_42

for recognizing foods. Kong and Tan [7] classified foods using multiple viewpoints. They compute SIFT and Gaussian region detector as the feature vector. Also, Kawano and Yanai [5] proposed a system which asks the user to draw a bounding box around food regions. Then, SURF based bag of features and color histograms are extracted and classified using a linear SVM. Matsuda *et al.* [9] proposed a method which takes into account the co-occurrence statistics of 100 food items. Similar to previous methods they applied Multiple Kernel Learning SVM on the image feature vectors such as color, SIFT, CSIFT, HOG and Gabor. Also, they utilized deformable part model, circle detector and JSEG methods for detecting candidate regions. Similarly, Hoashi *et al.* [4] classified 85 food classes by fusing BoF, color histogram, Gabor and HOG using Multiple Kernel Learning. In contrast to the previous methods, Yang *et al.* [13] classified food images with considering spatial relationship between food items. In this work, each image is represented by a pairwise feature distribution.

Lately, researchers have started to utilize Convolutional Neural Networks (ConvNets) in the task of food recognition. For instance, Christodoulidis *et al.* [1] proposed a 6 layer ConvNet to classify 7 items of food. They applied the ConvNet on the already segmented food images and used a voting method for determining the class of each food item. Also, Yanai and Kawano [12] fused Fisher Vector (FV) with pre-trained Deep Convolutional Neural Network features trained on 2000 ImageNet categories. Taking into account that food recognition systems might be implemented on mobile devices, we need a ConvNet with low memory and power consumption. Besides, time-to-completion of the ConvNet must be low in order to have a better user experience. To our knowledge, there are a few public food datasets such as UEC-Food100, UEC-Food256, and Pittsburgh Food Image Dataset. The problem of these datasets is that the number of samples in each class is scarce and highly imbalanced which makes them inapplicable for training a deep ConvNet with millions of parameters from scratch.

Contribution: In this paper, we partially address this problem by transferring knowledge of ConvNets trained on ImageNet dataset to a *smaller* ConvNet and fine-tune it using the dataset of food images. To be more specific, we first transfer knowledge of GoogleNet [11], AlexNet [8], VGGNet [10] and Microsoft Residual Net [2] on the UECFood 256 dataset. Our experiments show that if the knowledge of these ConvNet are transferred appropriately, they are able to outperform the state of art methods applied on this dataset. More importantly, we propose a method to transfer knowledge of the these ConvNets to a smaller ConvNet with less time-to-completion, less memory and similar accuracy.

2 Knowledge Transfer

One of the major barriers in utilizing ConvNets on the task of food recognition is that public food datasets are usually small. For this reason, it is not practical to train a ConvNet from scratch for this task. One alternative solution for solving this problem is to use the pre-trained ConvNets as a generic feature extraction method. For a ConvNet with L layers, we use

$\Phi_l(x; W_1 \dots W_l)$ to represent the vector function in the l^{th} layer parametrized by $\{W_1 \dots W_l\}$. With this formulation, $\Phi_L(x)$ represents the classification layer. Utilizing a ConvNet as a generic feature extractor means that we collect set $\mathcal{X} = \{(\Phi_{L-1}(x_1), y_1), \dots, (\Phi_{L-1}(x_N), y_N)\}$ where x_i is the image of food and y_i is its actual label. Then, we train a classifier (linear or non-linear) using the samples in $\mathcal{X}$.

As we show and explain in Sect. 3, this method does not produce accurate results with a linear classifier. For this reason, it is better to adjust the parameters of the ConvNet using the current dataset. By this way, the pre-trained ConvNets classify foods more accurately. As we mentioned earlier, we need an accurate ConvNet with lower time-to-completion and less memory requirement. Hence, we must find a way to compress these pre-trained ConvNet. To this end, we propose a method that transfers the knowledge of a large pre-trained ConvNets to a smaller ConvNet by keeping the accuracy high. Our method is inspired by the recently proposed method by Hinton *et al.* [3] called *Knowledge Distillation*. Given a pre-trained network $\mathbf{z}^{source} = \Phi^{source}(x; W_1 \dots W_{L_s})$, the aim of this method is to train $\mathbf{z}^{distilled} = \Phi_{distilled}(x; W_1 \dots W_{L_d})$ so that:

$$\| \frac{e^{z_i^{distill}}}{\sum_j e^{z_j^{distilled}}} - \frac{e^{\frac{z_i^{source}}{T}}}{\sum_j e^{\frac{z_j^{source}}{T}}} \| \tag{1}$$

is minimum for all sample in training set. In this equation, z_i indicates the i^{th} output and T is a parameter to soften the output of source network. One property of this method is that the classification score of the source ConvNet could be significantly different from the distilled ConvNet. This is due to the fact that there could be infinite combinations of classification score $z^{distill}$ to produce the same $softmax(z^{source})$ where $\|z^{distill} - z^{source}\|$ might be very large. For example, suppose that $softmax(z^{source}) = [0.99, 0.01]$ for a network with two outputs. Then, $z^{distill} = [10, 5.4049]$[1] and $z^{distill} = [100, 95.4049]$[2] will be the same $softmax(z^{distill})$. In other words, $\Phi_{distilled}(x)$ found by minimizing (1) may not accurately approximate $\Phi^{source}(x)$. Instead, it may mimic the normalized output of this function.

The advantage of this property is on distilled networks which are shallower than the source network. To be more specific, shallower networks *might* not be able to accurately approximate the z^{source} if they are not adequately wide. However, they might be able to produce $z^{distill}$ such that (1) is minimized. A drawback of this property is on networks that are deeper than or as deep as the source network. These networks might be able to accurately approximate z^{source}. However, training the distilled network by minimizing (1) is likely not to accurately approximate z^{source}. Besides, two different initialization might end up with two different distilled network where their $z^{distill}$ are significantly different from each other.

[1] [10,5.4048801498654111] to be exact.
[2] [100, 95.404880149865406] to be exact.

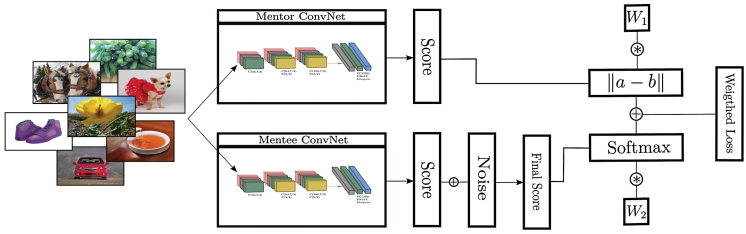

Fig. 1. Our proposed method for transferring knowledge from a larger network called Mentor to a smaller network called Mentee.

2.1 Proposed Method

We formulate knowledge transfer from one network to another network in terms of *function approximation*. Our proposed method is illustrated in Fig. 1. It consists of a Mentor ConvNet which is a pre-trained network and a Mentee ConvNet which is smaller and faster than the Mentor. Our aim is that Mentee performs similar to Mentor. Representing the Mentor with $\Phi_{mentor}(x)$ and the Mentee with $\Phi_{mentee}(x)$, we want to train $\Phi_{mentee}(x)$ such that $\forall_{x \in \mathcal{X}} \Phi_{mentor}(x) = \Phi_{mentee}(x)$ where $\mathcal{X}$ is a set consists of many *unlabelled* images. In other words, we formulate the knowledge transfer from Mentor to Mentee as a function approximation problem. By this way, the Mentee ConvNet is trained to approximate *un-normalized* Mentor ConvNet. Formally, our objective function is defined as *sum of square error*:

$$E = \sum_{i}^{N} \|\Phi^{mentor}(x) - \Phi^{mentee}(x)\|^2. \tag{2}$$

Theoretically, we do not need labelled images to transfer knowledge from Mentor to Mentee since the above loss function does not depend on labels of image. Consequently, we can use any large dataset of unlabelled images to approximate $\Phi_{mentor}(x)$ using $\Phi_{mentee}(x)$. By this way, Mentee is trained with non-food images. Notwithstanding, $\Phi_{mentee}(x)$ requires a large dataset of unlabelled images to be generalized. Since collecting this dataset is not tedious, we modified the above loss function as a weighted average of sum of square error and log likelihood:

$$E = \sum_{i}^{N} \alpha_1 \|\Phi^{mentor}(x) - \Phi^{mentee}(x)\|^2 - \alpha_2 \sum_{\forall\{i|y_i>0\}} \log(softmax(\Phi^{mentee}(x_i))). \tag{3}$$

During the first iterations, we set $alpha_2 = \epsilon$ so Mentee is mainly trained using sum of square error. Eventually, $alpha_2$ is increased in order to take into account the information coming from labelled images. In the next section, we explain the architecture of Mentor as well as Mentee networks.

3 Experiments

We transferred knowledge of AlextNet [8], GoogleNet [11], VGGNet [10] and ResNet [2] on UECFood256 dataset [6]. As we show shortly, if the knowledge of these ConvNets are properly adjusted to the domain of foods, they are able to outperform state-of-art methods. We also show that all of these ConvNets produce comparable results. However, taking into account their required memory and time-to-completion, GoogleNet is preferable over other ConvNets. For this reason, we use GoogleNet as the Mentor in Fig. 1.

Our aim is to train Mentee so it approximates Mentor as accurate as possible. For this reason, we choose the architecture of Mentee to be exactly similar to GoogleNet. However, we reduce the width of Mentee by reducing the number of filters in each inception module to 90 % of the original size. We use a combination of the ImageNet and the Caltech 256 datasets by ignoring their labels and use them as the set of unlabelled samples. Besides, we use UECFood256 dataset in order to compute the second term in (3) using labelled samples.

Results: In order to adapt knowledge of the ConvNets we mentioned earlier, we conducted the following procedure. First, all the layers are frozen except the last fully connected layer. Freezing a layer means that we set the learning rate of that particular later to zero so it does not change during backpropagation. Then, the last layer is trained on the food dataset. Second, we unfreeze the last two layers and keep the rest of the layers frozen. Third, the last three layers are unfrozen and the rest of the layers are kept frozen. Table 1 shows the top-1 and top-5 accuracies of the ConvNets in these settings.

Table 1. Adapting knowledge of ConvNets trained on ImageNet dataset to UECFood-256 dataset in different settings.

	Last layer		2nd last layer		3rd last layer	
	top-1 (%)	top-5 (%)	top-1 (%)	top-5 (%)	top-1 (%)	top-5 (%)
Alexnet	49	76	56	81	59	83
Googlenet	55	81	61	86	62	86
Vggnet	51	78	60	84	62	86
Resnet	60	83	62	86	NA	NA

The results suggest that adapting knowledge of the ConvNets must start from the two last layers. When we only adapt the knowledge of the last layer on UECFood-256 dataset, this means that the weights of the linear classifier are adapted. However, because these ConvNets are trained on ImageNet dataset their domain are different from UECFood-256 dataset. In other words, these ConvNets have been basically trained to distinguish the objects in ImageNet dataset. So, when they are applied on UECFood-256 dataset, foods might not be linearly separable in the last 2nd layer. Nonetheless, when the ConvNets are

adapted starting from the last two layers, they learn to transform the feature vectors produced in the last 3rd layer to be linearly separable in the last 2nd layer. Therefore, foods become linearly separable in the last layer. Besides, we observe that adapting the ConvNets starting from the last 3rd layer does not change the results. This might be due to the size of UECFood-256 dataset being small. Since the number of parameters starting from the last 3rd layer are high, they are not able to generalize properly provided by a small dataset.

Next, we use the GoogleNet adapted from the last three layers as the Mentor and trained the Mentee network. The architecture of the Mentee network has been explained in the beginning of this section. Table 2 illustrates the accuracy of Mentee for different valued of k. Comparing the plot with Table 1 shows that Mentee has accurately approximated the Mentor network and yet it is smaller and faster than Mentor.

We also compared our Mentee with the best results reported on UECFood-256 dataset. It is worth mentioning that [12] have used AlexNet as feature extractor and trained a classifier on top of it. Also, they have not augmented the original dataset. For this reason, their result is different from our result. In addition, DCNN-Food is modified version (number of neuron in the fully connected layer has been increased to 6144) of AlexNet which is specifically trained on the food images from ImageNet dataset. We observe that our Mentee has produced comparable results with respect to FV+DCNN-Food method with a much smaller network. Also, the top-5 accuracy of both of these methods are equal. Moreover, FV+DCNN-Food needs much more computations since it must compute Spatial Pyramid Fisher Vectors and apply a large network on the image. However,

Table 2. Accuracy of Mentee computed for different k

top (%)									
1	2	3	4	5	6	7	8	9	10
62	74	80	83	86	88	89	90	91	92

Table 3. Comparing our Mentee network with other methods reported in [12]

Method	top-1 (%)	top-5 (%)
Color FV	42	64
RootHOG FV	36	59
FV (Color+HOG)	53	76
DCNN	44	71
DCNN-Food	59	83
FV+DCNN	59	82
FV+DCNN-Food	64	86
Our Mentee	62	86

because we have trained our Mentor network properly (we trained the last three layers), the Mentee network is also able to predict classes, accurately (Table 3).

We have also computed the precision and recall of each class separately. You can find these results in the supplementary materials.

4 Conclusion

In this paper, we proposed a method for transferring knowledge from a bigger network called Mentor to a smaller network called Mentee in two phases. In the first phase Mentee uses unlabelled images to approximate the score produced by Mentor. In the second, phase, a dataset of labelled images are used to further tune the knowledge of small network in a supervised fashion. Our experiments on UECFood-256 dataset shows that pretrained ConvNet produce more accurate results when their knowledge is adapted starting from the last 2nd or 3rd layer. Using this information, we used GoogleNet as the Mentor and its compressed version as Mentee and transferred knowledge of the Mentor to Mentee. We showed that the Mentee network is as accurate as Mentor network and, yet, it is faster and consume less memory since its widths is less than the Mentor network.

References

1. Christodoulidis, S., Anthimopoulos, M., Mougiakakou, S.: Food recognition for dietary assessment using deep convolutional neural networks. In: Murino, V., Puppo, E., Sona, D., Cristani, M., Sansone, C. (eds.) ICIAP 2015. LNCS, vol. 9281, pp. 458–465. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23222-5_56
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1506.01497 (2015)
3. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS 2014 Deep Learning Workshop, pp. 1–9 (2015)
4. Hoashi, H., Joutou, T., Yanai, K.: Image recognition of 85 food categories by feature fusion. In: Proceedings - 2010 IEEE International Symposium on Multimedia, ISM 2010, pp. 296–301 (2010)
5. Kawano, Y., Yanai, K.: Real-time mobile food recognition system. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–7 (2013)
6. Kawano, Y., Yanai, K.: Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8927, pp. 3–17. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16199-0_1
7. Kong, F., Tan, J.: DietCam: regular shape food recognition with a camera phone. In: Proceedings - 2011 International Conference on Body Sensor Networks, BSN 2011, pp. 127–132 (2011)
8. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105. Curran Associates Inc. (2012)

9. Matsuda, Y., Hoashi, H., Yanai, K.: Multiple-food recognition considering co-occurrence employing manifold ranking. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2017–2020 (2012)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representation (ICLR), pp. 1–13 (2015)
11. Szegedy, C., Reed, S., Sermanet, P., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. arXiv preprint arXiv:1409.4842, pp. 1–12 (2014)
12. Yanai, K., Kawano, Y.: Food image recognition using deep convolutional network with pre-training and fine-tuning. In: 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW), pp. 1–6, June 2015
13. Yang, S., Chen, M., Pomerleau, D., Sukthankar, R.: Food recognition using statistics of pairwise local features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2249–2256 (2010)