Deep-based Openset Classification Technique and its Application in Novel Food Categories Recognition

Jakramate Bootkrajang, Jakarin Chawachat, and Eakkapap Trakulsanguan

Department of Computer Science, Chiang Mai University, Thailand
{jakramate.b, jakarin.c, eakkapap.t}@cmu.ac.th

Abstract. Being able to accurately recognise food categories from input images has many possibly useful applications such as content-based recipe searching or automatic intake calories tracking. Convolutional neural networks has been successfully applied in a number of food recognition tasks. Despite its impressive predictive performance on closed datasets, there is currently no standard mechanism for distinguishing unknown object classes from the known ones leading to invalid classification attempts even on non-food images. In this paper, we study a technique for detecting whether input images are beyond the scope of CNN's knowledge. The idea is to model the final activation vectors of data from the known classes using a data description method namely the support vector data description. We can then reject network's prediction if the activation vector of the query image is too different from the known ones as generalised by the model. Experimental results on a subset of UEC-FOOD100 datasets demonstrated that the proposed method was able to accurately classify instances from the known classes while also being able to satisfactorily reject the prediction of novel food image compared to two commonly used baselines.

Keywords: openset classification · food recognition · deep learning.

1 Introduction

Food categories recognition is one of the interesting topics among several visual recognition tasks. An accurate food recogniser can be employed in many real world applications such as intake calories estimation [12], dietary assessment [13] or recipe searching from image [2], to name just a few. Several attempts had been made to tackle the problem of food categories classification problem. The majority of previous research was based on Support Vector Machine using specially crafted visual features [16, 5]. Some classical pattern recognition techniques such as k-nearest neighbour has also been employed for the task [7]. Recently, Convolutional Neural Networks (CNN) is gaining more popularity due to its impressive performance in visual recognition tasks [6, 18]. Unlike previous image classification approaches which rely on the quality of visual features extracted from the image, CNN learns good feature representation simultaneously with learning the classifier. The model has been adopted for food recognition task [11, 10, 20] and has been shown to outperform existing approaches.

Despite of its impressive predictive performance, CNN and in fact any classifier in general still has some limitation for real-world usage. In particular, classical supervised learning assumes query input comes from the same data distribution as the one used to train the model. In the context of food categories recognition, we implicitly assume that query food image be one of the known food categories. Unfortunately, this assumption does not always hold true in real-world recogniser deployment. Surely, in such case, the classification model still make a prediction even though the query image is not a food image or is a new type of food.

The aforementioned limitation motivates us to study the problem of detecting whether the prediction should be made for incoming query image. We would want our recognition model to be able to reject the prediction for input image which does not belong to the known classes while still gives accurate prediction for those that are within the classification scope. The problem of this kind is not new and it has been studied in the past under the name *novelty detection* [15] (and sometime interchangeably as outlier detection or anomaly detection). From machine learning perspective, novelty detection problem can be approached with one-class classification [14] where there are already many learning algorithms available. Recently, the problem is increasingly known in the machine learning community as an openset classification problem [17, 9]. There is, however, one subtle difference between one-class classification and openset classification problem. In one-class classification, the task is to differentiate between a *single* target class and other possible inputs. Meanwhile, in openset classification, we want to differentiate *multiple* known classes from other possibilities. The challenge is that the set of known classes might not form an obvious single class and it is interesting if existing one-class classification technique will work in this more challenging case.

To the best of our knowledge, there were not many attempts to studying openset classification problem within the scope of convolutional neural network. A seemingly straightforward mechanism for detecting if an input is beyond the scope of training data is by means of *class posterior thresholding*, where a prediction would not be made if the posterior probability of the most probable class is below some predefined threshold. A more advanced approach could involve the modelling of the activation vector, e.g., the output of the final layer of the network. The approach is based on the observation that inputs that belong to the same object class should have similar activation vector. Therefore, if we can summarise and construct representative activation vector either globally or locally (i.e., having one representative for each of the data classes), we should be able to detect if the query image actually belongs to one of the known classes by comparing their activation vectors. The work in [1] took this route and modelled the representative using the mean activation vector computed over correctly classified examples in each class. The prediction is rejected if the distance of input's AV is too far from the mean AV.

We shall also take this route but extend the data description model into a more complex one by using the Support Vector Data Description. Hopefully, the complex data boundary creates by the SVDD might be beneficial. Further, our work differ from [1] in that their analysis was based on activation values from the penultimate layer of the network while our approach models the final activation vectors directly. We believe that the relatively lower dimensional nature of the final activation vectors, which scales with the number of classes might better suit our modelling choice.

The rest of the paper is organised as followed. Background and the details of the proposed SVDD-based novel instance detection approach are presented in Section 2. Section 3 presents the empirical studies and discussion of the results. Section 4 presents the concluding remarks and outlines future research direction.

2 Background and Methods

Formally, constructing a classifier is a task of inferring a function $f : X \to Y$, which maps instances in a feature set X to an instance in label set Y using a subset of examples in the form of $(\mathbf{x}_i, y_i)_{i=1}^N$ pairs independently and identically (i.i.d) drawn from the joint distribution $D : X \times Y$. The goal of the learning is to be able to use the resulting classifier $f(\cdot)$ for assigning $y \in Y$ to an unseen query instance $\mathbf{x}_q$ from X with high accuracy. In classical setup, we implicitly assume that the query point is also from X. However, such assumption is quite unrealistic for real world classification model deployment where there is little guarantee that the query input will be one of the instance in X. And we would like to detect when this happens.

Our approach for detecting whether $\mathbf{x}_q$ is beyond the classification scope relies on the analysis of the final activation vector: the vector of output from the final layer of the neural network $f(\cdot)$. We denote the activation vector of an input image $\mathbf{x}_i$ by $f(\mathbf{x}_i) = \mathbf{v}_i = [v^1, \ldots, v^K]$. Usually, given a classification task of K target classes, there will be K output nodes. Accordingly the activation vector is a real-valued vector in $\Re^K$. Often, the output values from the final layer is normalised such that $\sum_{j=1}^K v^j = 1$.

2.1 Convolutional Neural Networks

Before proceeding, we would like to first outline the architectures of the deep neural networks employed. A convolutional neural network is a classifier which can be divided into two parts: the convolutional layers and the fully connected layers. The convolutional layers part is responsible for extracting visual features from the images while the fully connected layers takes visual features extracted by the convolutional layers and assigns class label to the image. Various CNN architectures published to date differ from each other primarily at the convolutional layers level. It has been empirically shown that abstract visual features (especially those found at the very first convolutional layers) are high level features and are shared among various kind of visual recognition tasks [21]. In practice, the weights of the convolutional layers can be transferred from some pretrained networks of which the weights were sufficiently learnt from massive visual datasets and should provide a good starting point for further fine tuning. In this work we employed four well-known deep convolutional architectures trained on ImageNet dataset, namely VGG16 and VGG19[18], ResNet50 [6] and DenseNet121 [8]. The weights of the convolutional layers were transferred and were frozen during training and only the weights of the fully connected layer was trained via the standard back-propagation methodology. Our fully connected network is composed of an input layer with 256 nodes, a hidden layer with 64 nodes and an output layer with 20 nodes. The weights from input to hidden and hidden to output layer are subjected to 0.3 dropout rate. Activation functions in the input and hidden layers were the sigmoid while the softmax function is used in the output layer. We trained the fully-connected network using batch size of 16 and small learning rate of 10^{-5}.

2.2 Activation Vector Data Description model

To differentiate between activation vectors belonging to the set of known classes and those from the unknowns, we study the idea of employing the Support Vector Data Description (SVDD) [19] well used in the area of one-class classification to model activation vectors from the known classes. Intuitively, SVDD starts with a hypersphere of radius R centred at $\mathbf{a}$. The objective is to find a hypersphere with minimum radius which also encloses all of the data. Formally, the objective of SVDD is to minimise the following loss function: $L(R, \mathbf{a}) = R^2$ and subject to $||\mathbf{v}_i - \mathbf{a}||^2 \leq R^2, \forall i$. The above formulation is rather rigid such that requiring all $\mathbf{v}_i$ to lie in the hypershere could be difficult in real world usage where some outliers exist. To mitigate the problem, *slack variable* $\xi_i \geq 0$ could be introduced into the formulation, yielding an objective function in its primal form: $L_{primal}(R, \mathbf{a}) = R^2 + C\sum_i \xi_i$ subject to $||\mathbf{v}_i - \mathbf{a}||^2 \leq R^2 + \xi_i, \forall i$. Here C is the hyperparameter which controls the trade-off between the volume of the hypersphere and errors. The new formulation expresses the fact that *almost* all data (the activation vectors in our case) is required to fall within the hypersphere. Following [19], minimising the primal is equivalent to maximising its dual form:

$$L_{dual}(\alpha) = \sum_i \alpha_i \kappa(\mathbf{v}_i, \mathbf{v}_j) - \sum_{i,j} \alpha_i \alpha_j \kappa(\mathbf{v}_i, \mathbf{v}_j) \tag{1}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i = 1$. Here, $\kappa(\cdot, \cdot)$ is a positive definite reproducing kernel function that enables the construction of non-linear data boundary. The activation vectors $\mathbf{v}_i$ where its corresponding $\alpha_i > 0$ are called the Support Vectors (SVs) for the description. In this work, we will work with the Radial Basis Function kernel (RBF) given by $\kappa(\mathbf{v}_i, \mathbf{v}_j) = \exp(||\mathbf{v}_i - \mathbf{v}_j||^2)/2\sigma^2$. We used $\sigma = 2^{-4}$ throughout the experiments. Accordingly, a test image $\mathbf{x}_q$ is considered novel if the distance of its $\mathbf{v}_q$ from the SVs, given by

$$\sum_{i \in SVs} \alpha_i \exp\Big(\frac{||\mathbf{v}_q - \mathbf{v}_i||^2}{2\sigma^2}\Big), \tag{2}$$

is greater than some predefined threshold ρ. Since the value of $\mathbf{v}$ can be very small e.g., less than 10^{-10}, and can cause some numerical instability, we propose to work instead with the logarithm of $\mathbf{v}$. We shall refer to the method described above as Activation Vector Data Description (AVDD) to emphasise the modelling of the activation vectors using SVDD. Algorithm 1 summarises the steps to construct the AVDD while the steps for detecting novel instance are outlined in Algorithm 2.

Algorithm 1 The construction of the Activation Vector Data Description.

Input: A set of activation vectors of the training data $(\mathbf{v}_i)_{i=1}^{N}$
  Perform logarithmic transform $\tilde{\mathbf{v}} = \log \mathbf{v}$
  Construct SVDD model using Eq.(1) based on the transformed $(\tilde{\mathbf{v}}_i)_{i=1}^{N}$
Output: Optimised $(\alpha_i)_{i=1}^{N}$

Algorithm 2 The openset detection step.

Input: An activation vector of the test data $(\mathbf{v}_q)$ and parameters of AVDD $(\alpha_i)_{i=1}^{N}$
  Perform logarithmic transform $\tilde{\mathbf{v}}_q = \log \mathbf{v}_q$
  Calculate distance of $\tilde{\mathbf{v}}_q$ from the support vectors $(\tilde{\mathbf{v}}_i)_{i \in SVs}$ using Eq.(2).
  if distance $> \rho$ **then**
    $\hat{y}_q =$ "novel instance"
  else
    $\hat{y}_q = \arg\max_j \tilde{v}_q^j$
  end if
Output: $\hat{y}_q$

3 Empirical evaluations

We will now study the effectiveness of the proposed AVDD detection method in openset classification problem. The main question is how well the proposed method identify unknown input instance while also being able to recognise instances from target classes. We shall compare the detection performance of our proposed method with two commonly used baselines. The first baseline is the simplest mechanism for novelty detection. The scheme rejects the prediction of input $\mathbf{x}_q$ if the class posterior probability of the most probable class turns out to be less than some predefined threshold, e.g., $\max_j v_q^j < \theta$. We will refer to this method as Class Posterior Thresholding (CPT). The second baseline involves the calculation of Mean Activation Vectors (will be referred to hereafter as MAV method) for each class. The approach then rejects the prediction if the activation vector of the query $\mathbf{v}_q$ is too different from the mean activation vector of the predicted class, e.g., $dist(\mathbf{v}_q, \mu_{\hat{y}_q}) > \beta$. For simplicity, we considered standard Euclidean distance for similarity measurement.

Table 1. The datasets used in this study are divided into three groups. *FOOD20* is used to train the recognition model. *OPEN-FOOD* is used to test the capability of the model in detecting unknown but related objects. *OPEN-OBJECT* is a set of unrelated objects.

Dataset	Class labels
FOOD20 #instances 3338 #classes 20	rice, eels on rice, pilaf, sushi, chicken rice, fried rice, toast, croissant, roll bread, hamburger, pizza, sandwiches, udon noodles, spaghetti, Japanese pancake, takoyaki, gratin, cutlet curry, potato salad
OPEN-FOOD #instances 1396 #classes 10	chicken-and-egg on rice, pork cutlet on rice, beef curry, tempura bowl, bibimbap, raisin bread, chip butty, beef noodle, tensin noodle, fried noodle
OPEN-OBJECT #instances 301 #classes 10	apple, bird, car, carrot, cat, dog, doll, fish, orange, plane

3.1 Datasets and protocol

The food images used in this study are from the UECFOOD100 dataset [11]. The original dataset contains visual images of 100 Japanese food categories. Region Of Interest (ROI) information is provided for every image. Our preprocessing steps involve extracting food images according to the ROIs and resizing the image to 224×224 pixels to match the input requirement of the VGG16 network.

We randomly sampled 20 food classes from the dataset for our experiment. We will refer to this set of data as *FOOD20* dataset. To evaluate the novelty detection performance, we set apart another 10 classes from UECFOOD100, called *OPEN-FOOD* and another 10 classes of general objects images from Imagenet dataset [3] which are irrelevant to food called *OPEN-OBJECT*. Table 1 summarises the datasets used in this study.

3.2 Results: performance on known classes

We first want to establish a *closed set accuracy*. The accuracy is identical to the accuracy obtained in the idealised supervised learning scenario where testing data are from the same data distribution as that of the training data used to train the model. To do this, we randomly split *FOOD20* data into training and testing set using 90/10 percent ratio. We trained the models until they sufficiently converged on the training data. We then validated theirs performance on the remaining 10 percent of data and recorded the classification accuracies. We note that in this case the models are allowed to predict all of the testing examples without employing the novelty filtering mechanism. We repeated the aforementioned procedure for 10 repetitions in order to get reliable statistics. Table 2 reports the average top-1 and top-5 classification accuracies together with their standard errors of the four CNNs employed.

We observed that top-1 accuracies of all CNNs are well above 80% except for DenseNet121 which slightly lagged behind. In general, there seems to be

Table 2. Top-1 and Top-5 closed set accuracies of four convolutional neural networks employed in this study.

	VGG16	VGG19	ResNet50	DenseNet121
Top-1 accuracy	83.34 ± 1.48	82.68 ± 1.91	86.71 ± 1.45	76.66 ± 1.84
Top-5 accuracy	97.36 ± 0.07	96.61 ± 1.03	97.37 ± 1.21	94.61 ± 1.14

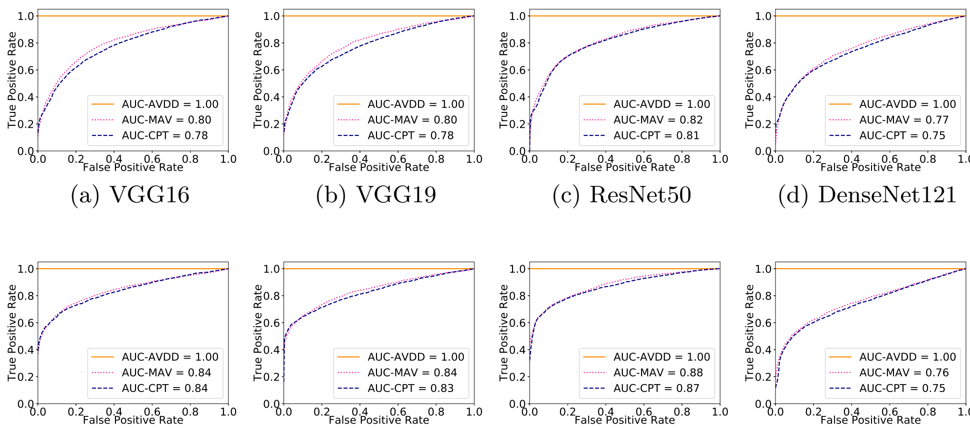
some confusion among similar kind of food e.g., rice-based dishes. Meanwhile, the top-5 performances from all models are quite impressive with accuracies exceeding 97%. Although, the top-5 performances are acceptable, we believe that more sophisticated technique can surely be incorporated into the classification model to enhance the top-1 predictive performance and we plan to do so in the subsequent work. The results are also inline with the results reported in the original paper of the dataset [10]. This suggests that the CNN architectures used in this study are deemed suitable for the task as all were capable of learning the regularities in the data to some extent. Next, we shall turn to study the effect of novelty detection mechanism on the performance of the chosen CNN models.

3.3 Results: novel classes detection

In this section we shall evaluate the proposed novelty detection mechanism. We would like to quantify the error that each of the comparing mechanisms makes during the detection process. There are two types of error: type 1 and type 2 error. Type 1 error occurs when the detector thinks that query image is from a novel class but in reality the query is from one of the known classes. Type 2 error occurs when the query image is indeed outside the classification scope but the detector thinks that it is not. A good way to summarise both errors graphically is by constructing a Receiver Operating Characteristic (ROC) curve [4]. We followed the same training protocol as described above but instead of predicting class labels during testing, we mixed the remaining 10 percent of *FOOD20* data held out for testing with data from *OPEN-FOOD* to get the first set of open data, and with *OPEN-OBJECT* to get the second open dataset. The task is then to tell whether images in the mixed testsets are the images from the held-out *FOOD20* or not. We repeated the experiment for 10 repetitions while recording True Positive Rate and False Positive Rate in each run. The average ROC curves of the three mechanisms combined with four respective CNNs are presented in Figure 1.

From the results, we notice that detecting unknown objects based on activation vectors, either by MAV approach or AVDD approach, is more effective than the standard CPT method. We also see that AVDD was better than both MAV and CPT methods by a large margin in both of the open datasets, partly thanks to its non-linear data boundary. We speculate that the similarities between some of the classes in *FOOD20* and *OPEN-FOOD*, e.g., chicken rice vs chicken-and-egg on rice, might contribute to the poor performance of CPT. Meanwhile, MAV method might not be delicate enough to differentiate between two different AVs

Fig. 1. The ROC curves for the three novelty detection mechanisms on OPEN-FOOD (top) and OPEN-OBJECT (bottom), together with their associated AUCs.



(a) VGG16 (b) VGG19 (c) ResNet50 (d) DenseNet121

(e) VGG16 (f) VGG19 (g) ResNet50 (h) DenseNet121

that happened to have the same Euclidean distance from the mean. This suggests that the proposed mechanism is quite promising in detecting novel classes for various CNN-based food classification models in real-world usage.

The ROC curve summarises the detection performance at various thresholding values. Still, one question remains unanswered namely, how do we choose the cutoff threshold sensibly? It is somehow unrealistic to assume there exists open validation set for threshold selection as by definition the possibilities of instances in the openset are endless. For the posterior thresholding baseline, i.e., CPT in this study, the threshold reflects our requirement for classifier's confidence which varies from task to task. For less sensitive task we could, for example, aim for $\theta \approx 0.9$ while in more critical situation we might want to set θ a little bit higher. For MAV method, the determination of cutoff threshold is less straightforward as the distance measure lacks probabilistic semantics, and we think this is one of the difficulty associated with this kind of approach. Interestingly, for AVDD, we observed that the distances of AVs of novel points from the support vectors mostly concentrate at some value. This allows us to select good cutoff threshold using only a small set of open data.

We then investigated whether or not input instances from *OPEN-OBJECT* dataset can be used to facilitate the selection of ρ, the cutoff threshold. According to the concentration phenomenon mentioned earlier, we set ρ to be the distance of an AV of a random example from *OPEN-OBJECT* from the support vectors, minus some small number e.g., 0.0001. This is to compensate the tiny variance associated with the distances of AVs of other instances from *OPEN-OBJECT* dataset. The heuristic is sensible for two reasons. First we did not assume the availability of novel food images since if we have such data we could have already

Table 3. Top-1 and Top-5 open set accuracies of the convolutional neural network paired with each respective detection mechanisms on *FOOD20 + OPEN-FOOD* dataset. Closed set performances are included for reference.

Methods	Top-1	Top-5	*OPEN-FOOD* recognition rate
VGG16 + AVDD	**83.34 ± 1.48**	**97.36 ± 0.07**	**99.78 ± 0.07**
VGG16 + MAV	73.73 ± 5.55	79.76 ± 8.12	58.08 ± 16.17
VGG16 + CPT	66.46 ± 1.85	69.04 ± 2.31	72.06 ± 1.11
VGG16 (closed set)	83.34 ± 1.48	97.36 ± 0.07	N/A
ResNet50 + AVDD	82.51 ± 2.35	89.06 ± 2.71	**100.00 ± 0.00**
ResNet50 + MAV	76.54 ± 3.82	79.72 ± 4.74	63.62 ± 9.55
ResNet50 + CPT	83.05 ± 1.82	89.82 ± 1.47	36.48 ± 1.44
ResNet50 (closed set)	86.71 ± 1.45	97.37 ± 1.21	N/A

included it in the training set. Second, *OPEN-OBJECT* is publicly available and can be obtained quite easily without additional overhead. We adopted the same mechanism for setting β, the threshold for MAV. The cutoff thresholds of CPT was simply set to $\theta = 0.9$. We then evaluate the threshold selection heuristic by measuring an *open set accuracy* which is the ratio of instances, which were not caught by the detection mechanism and were also correctly classified, over the total number of test instances in *FOOD20*. Due to page limit, we present the open set performances of the three detection methods combined with VGG16 and ResNet50 using *FOOD20 + OPEN-FOOD* dataset in Table 3.

From the results, we see that all detection mechanisms incurred a slight drop in both top-1 and top-5 accuracies. This is expected though because some legitimate predictions might have been discarded by the detection mechanisms. Still, we observe that AVDD, among the three methods, was able to retain the top-1 and top-5 accuracies better while also being effective in detecting novel inputs. The results also validated the usefulness of the threshold selection heuristic and hinted that VGG16 + AVDD might be a good pair for the task.

4 Conclusions

We studied a novel food categories recognition for convolutional neural networks. Our method relies on the construction of data description model by means of the support vector data description. The model rejects the prediction and alerts that the input is from unknown food class if the activation vector of the query image is too different from the model. The empirical study was setup to compare the effectiveness of the proposed method with the traditional baselines of class posterior thresholding and mean activation vector methods. The detection capability of the proposed method was shown to be promising. What remains unexplored in this work is how to further make use of the novel objects. One possibility is to combine a so-called self learning methodology to query similar images from image search engine while extracting label from the most probable image tags and use the new information to retrain the recognition model.

Acknowledgement

The research is supported by the Faculty of Science, Chiang Mai University.

References

1. Bendale, A., Boult, T.E.: Towards open set deep networks. CoRR **abs/1511.06233** (2015)
2. Chen, J., Ngo, C.W.: Deep-based ingredient recognition for cooking recipe retrieval. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 32–41 (2016)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. of IEEE CVPR. pp. 248–255 (2009)
4. Fawcett, T.: An introduction to roc analysis. Pat.Rec.Lett. **27**(8), 861–874 (2006)
5. He, H., Kong, F., Tan, J.: Dietcam: Multiview food recognition using a multikernel svm. IEEE Journal of Biomedical and Health Informatics **20**, 848–855 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of IEEE CVPR. pp. 770–778 (2016)
7. He, Y., Xu, C., Khanna, N., Boushey, C.J., Delp, E.J.: Analysis of food images: Features and classification. In: Proc. IEEE Image Processing. pp. 2744–2748 (2014)
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proc. of IEEE CVPR. pp. 4700–4708 (2017)
9. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: ECCV. pp. 393–409. Springer (2014)
10. Kawano, Y., Yanai, K.: Foodcam: A real-time food recognition system on a smartphone. Multimedia Tools and Applications (2014)
11. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In: Proc. of IEEE ICME (2012)
12. Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., Murphy, K.P.: Im2calories: towards an automated mobile vision food diary. In: Proc. of IEEE Int. Conf. on Computer Vision. pp. 1233–1241 (2015)
13. Mezgec, S., Koroušić Seljak, B.: Nutrinet: A deep learning food and drink image recognition system for dietary assessment. Nutrients **9**(7), 657 (2017)
14. Moya, M.M., Koch, M.W., Hostetler, L.D.: One-class classifier networks for target recognition applications. NASA STI/Recon Technical Report **93** (1993)
15. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. Signal Processing **99**, 215–249 (2014)
16. Pouladzadeh, P., Villalobos, G., Almaghrabi, R., Shirmohammadi, S.: A novel svm based food recognition method for calorie measurement applications. In: Proc. of IEEE ICME (Workshops). pp. 495–498 (July 2012)
17. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE TPAMI **35**(7), 1757–1772 (2013)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Tax, D.M., Duin, R.P.: Support vector data description. Machine Learning **54**(1), 45–66 (Jan 2004)
20. Yanai, K., Kawano, Y.: Food image recognition using deep convolutional network with pre-training and fine-tuning. In: Proc. of IEEE Int. Conf. on Multimedia & Expo Workshops. pp. 1–6 (2015)
21. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NIPS. pp. 3320–3328 (2014)