Food Recognition in the Presence of Label Noise

Ioannis Papathanail[1], Ya Lu[1], Arindam Ghosh[2], Stavroula Mougiakakou[1]

[1] ARTORG Center for Biomedical Engineering Research, University of Bern, Bern, Switzerland
[2] Oviva S.A., Zurich, Switzerland

Abstract. The scope of multi-label image classification is to recognize several objects that appear within a single image. In the current paper we consider the task of multi-label food recognition, where the labels of the images in the training set are noisy, as they are annotated by inexperienced annotators. In our approach, a proposed noise adaptation layer is appended to a pretrained baseline model, aiming to correctly learn from these noisy labels. From the baseline model, predictions are made on the training set, given the images and the noisy labels. Out of these predictions and the noisy labels a confusion matrix is being created. This confusion matrix is used to initialize the weights of the noise layer and the full model is retrained on the training set. The final predictions for the testing set are made from the baseline model, after its weights have been readjusted by the noise layer. We show that the final model significantly increases the performance on noisy datasets.

Keywords: Multi-label image classification, noisy data.

1 Introduction

Recent estimations of the World Health Organisation (WHO) [1], show that more than 1.9 billion and 650 million people worldwide live with excess adiposity and clinical obesity respectively. In addition to that, around 422 million people live with diabetes. According to the American Centers for Disease Control and Prevention (CDC) [2], a healthy diet can not only prevent overweight and obesity, but also lower the risk of numerous chronic diseases such as type 2 diabetes, heart disease and some forms of cancer. In order for a person to adhere to a healthy diet, monitoring of their dietary habits, and, thus, proper dietary assessment is necessary.

Continuous dietary assessment has traditionally been performed using instruments such as food diaries and 24-hour dietary recall [3]. One major challenge of such subjective assessment methods is self-reporting errors due to the inability to estimate the correct portion sizes. Nowadays, with the development of Artificial Intelligence (AI) and computer vision, dietary assessment can be done automatically, with high accuracy and efficiency, using a smartphone camera. Several methods have been proposed for the automatic estimation of the nutritional content of an image [4]-[7] and the procedure usually consists of the following three steps i) food recognition ii) food segmentation and iii) volume estimation & nutrient content calculation.

Food recognition is the fundamental step of the dietary assessment. At the early stages, hand-engineered features and traditional image classifiers were used for food recognition [8]-[10], while recently the use of deep learning algorithms have significantly improved the accuracy of food recognition tasks [4]-[6] [11].

Most of the existing approaches in the domain of food recognition focus on the single-food recognition task, i.e. each input image corresponds only to a single food label [12] [13]. However, in a real scenario it is common for a food image to contain more than one food labels.

In the field of multi-label food recognition, the image is first segmented into parts that contain a single food category, followed by a classification of the individual segments [11] [14] [15]. Although these methods can yield satisfactory results, they require additional computation time for food segmentation. In [14] and [16] a "sigmoid" layer is applied at the end of a classification network, in order to predict the multi-label food categories that appear in an image. Even though these methods tend to have better results, they depend on large-scale databases [10] [17] with pure annotations. Collecting expert label data with pure annotations is an extremely time-consuming task and low inter-annotator agreement is a fundamental characteristic of any such task.

In this paper, we propose a simple but effective approach in order to deal with noisy labels in the training set in the case of the multi-label food recognition problem. Specifically, we build a Confusion Matrix (CM) that represents the label noise distribution of the training set. The CM is used to initialize the weights of a Noise Layer that connects the correct labels with the noisy ones and it is removed afterwards, so that predictions can be made on a clean testing set. Therefore, the Noise Layer does not have any negative effect on the computation time. The dataset that is used in our method contains images from real end-users, making it easily applicable to real-world problems. Finally, the training set does not have a subset of images with clean labels, that could possibly help the training process.

2 Related Work

2.1 Food Recognition

In [8], the proposed method first detects candidate regions that probably include foods inside them and then estimate the probability of each region belonging to every food category. A similar approach is used in [9] and [10], where the Bag of Features model was adopted to represent an image as a collection of local features. These methods make use of hand-engineered features like color histograms, Scale Invariant Feature Transforms or a combination of them [18] and simple architectures, like Support Vector Machines, in order to classify the images. More recently, with the advance of Deep Learning, the use of more complicated architectures like Convolutional Neural Networks (CNN) for food recognition [4]-[6] [11] tends to outperform the above approaches. These methods often use networks like GoogleNet [19], ResNet [20] or InceptionV3 [21] that are pretrained on large image datasets like ImageNet which contains 1.2 million images with 1000 classes. These networks are then fine-tuned for the food recognition task.

The methods reviewed in this Section depend on large databases, like the UEC-FOOD100 [8] [10] or the UEC-FOOD256 [22] where the labels of the images are free of noise. In practice, relying on human experts to annotate a dataset that does not contain label noise can prove costly and slow. Although the Food-101 dataset [17] contains some label noise, it contains 101,100 images divided equally into 101 classes. Therefore, the dataset can be used only for single-food recognition and, also, has the same number of samples for each category; thus, making the training process easier. In this paper we use a dataset that contains images from real end-users, and, therefore, contains label noise, has imbalanced classes and each image can contain multiple food categories.

2.2 Noisy labels

The presence of noise in the labels of the training set can heavily influence the results of food recognition. Zhang *et al.* [23] showed that deep neural networks can overfit on the noisy labels and generalize poorly on a clean testing set. An in-depth survey has been conducted in 2013 [24], regarding the different types of noise, the effects of the label noise and different methods of dealing with label noise, such as: label noise-cleansing, noise-robust methods or algorithms that try to model label noise during training. These methods are often used in tandem to yield better results.

The label noise-cleansing method aims to improve the quality of the data by either relabeling the samples that are likely to be mislabeled [25] [26], pruning them [27] or applying sample weights to the examples, based on the likelihood that their labels are correct [28] [29]. In [30] and [31] a small set of clean samples is used in conjunction with a much larger dataset that contains noisy labels, in order to assist in the training process. A method called “curriculum learning” seems to gain more and more success in the fields of learning from noisy data [28] [32], based on the idea that networks can benefit if they start learning with easy examples and progressively move to more complex ones. However, the above techniques either assume that there is a subset that contains clean labels, can sometimes discard useful data or adopt a complex method that can increase the computational time.

Other methods propose building models that are robust to label noise [33] [34]. Natarajan *et al.* [33], provided a way to modify a given loss function for binary classification, so that it is more robust to label noise. In [34] the CNN learns visual features by being trained on millions of weakly-labeled images. Nevertheless, the label noise is not actually being considered in these cases.

Finally, there are methods that try to estimate the noise transition matrix between the noisy labels and the true, hidden ones. Sukhbaatar *et al.* [35] suggested appending a linear layer on top of the baseline CNN, that can be interpreted as the noise-transition matrix. However, in their method, the noise depends only on the true labels and not the images themselves. In [36], a similar approach is used, but the output of the baseline CNN is fully connected to the noisy-label layer, and the noise depends also on the image features.

Our approach is similar to that of [36], but we extend the problem for the case of multi-label classification. Moreover, the noisy labels in the training set are not hard assigned to classes but they have a probability of belonging to each class, equal to the average of the annotations done by the different annotators. This method, compared to others [30] [31], does not rely on a subset of clean labels that can assist in the training process and does not affect the computation time.

3 Method

Our goal is to train a multi-label classifier that can distinguish the food categories that appear in a single RGB image. The labels of the training dataset are noisy, in a sense that they are annotated by inexperienced annotators.

In our method, a baseline image classification model (BM) is trained first, in order to give some preliminary results. For the BM, any prevalent network architectures such as GoogleNet [19], InceptionV3 [20] or ResNet [21] can be used. Here, we used the ResNet-101 and the InceptionV3 as BM due to their good performance in image classification [37]. The BM is used to make predictions on the training set and, out of these predictions and the noisy labels, the CM is built. The CM is a simple, yet valid representation of the dataset's noise. On top of the BM, a Noise Layer is added, using the values of the CM as its weights, aiming to predict the noisy labels. The final predictions on a clean testing set are done by the model, after removing the Noise Layer, that was used to learn the noise distribution in the training set. The architecture of the full model (FM) is depicted in Figure 1.

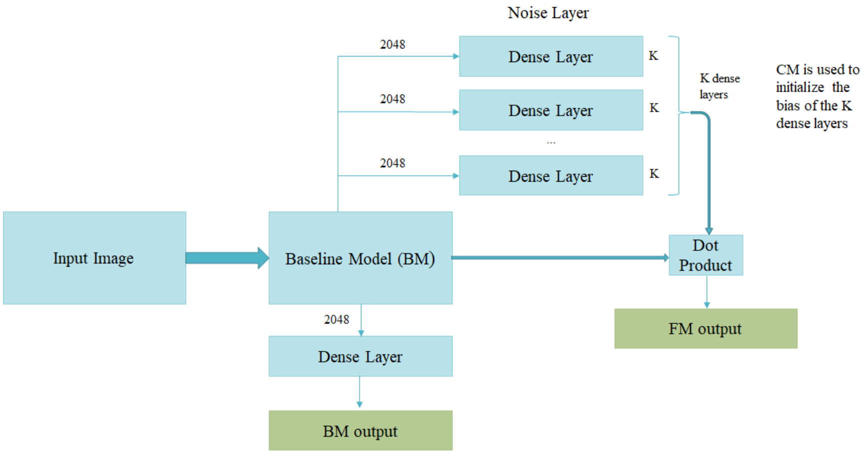

Fig. 1. The architecture of our full model (FM)

3.1 Confusion Matrix (CM) Building

Assume the training dataset contains N images that belong to one or more classes, out of K classes in total. Let $\boldsymbol{x_i} \in \mathrm{X} \subset R^d$ be the feature vector of the i^{th} image ($i \leq N$) and $\boldsymbol{z_i} = \{Z_{i1}, \dots, Z_{iK}\} \in [0,1]^K$, where Z_{ir} is the average of the annotations for the

i^{th} image and the r^{th} class. However, $\mathbf{z}$ is just a noisy version of the true, hidden labels $\mathbf{y}$, which are unknown.

At the first step, the BM is trained on the noisy dataset $D = \{(\boldsymbol{x_1}, \boldsymbol{z_1}), \dots, (\boldsymbol{x_N}, \boldsymbol{z_N})\}$. Instead of using the BM to make predictions on the testing set, it is used to predict the labels of the training set. The predictions are $\boldsymbol{P} = \{\boldsymbol{p_1}, \dots, \boldsymbol{p_N}\}$, where $\boldsymbol{p_i} = \{P_{i1}, \dots, P_{iK}\} \in [0,1]^K$ are the probabilities that image i contains labels 1 to K. By doing that, we can observe which classes are confidently assigned to the annotated classes and which differ from them, implying they are correctly or probably incorrectly annotated, respectively.

From the predictions $\boldsymbol{P}$ and the noisy annotations $\boldsymbol{Z} = \{\boldsymbol{z_1}, \dots, \boldsymbol{z_N}\}$, a $CM \in R^{K \times K}$ is built, considering also that the problem is multi-label. In addition to that, the noisy labels are the probabilities of every image belonging to each class, which is equal to the average of the annotations. The rows of the CM depict the predictions of the BM (that can be treated as an estimation of the true hidden labels $\mathbf{y}$) and the columns depict the noisy labels. For each pair $(\boldsymbol{p_i}, \boldsymbol{z_i})$ the CM is being updated as described below.

Initially, the classes that are apparent in both $\boldsymbol{p_i}$ and $\boldsymbol{z_i}$ are found. If, for a class α ∈ K, $P_{i\alpha} > th_\alpha$ and $Z_{i\alpha} > 0$, then in the α^{th} row and α^{th} column of the CM, the value $Z_{i\alpha}$ is added. The threshold for predicting class α, th_α, is calculated so that the number of images with $P_{ia} > th_a$ is equal to the number of images with $Z_{ia} > 0$. In other words, if A is the number of images that at least one annotator has assumed to contain class α, then the A images with the highest predicted probabilities for class α are assigned to the class α.

Assuming μ classes appear in $\boldsymbol{z_i}$ but not in $\boldsymbol{p_i}$ and ν classes appear in $\boldsymbol{p_i}$ but not in $\boldsymbol{z_i}$, if $P_{i\beta} > th_\beta$ and $Z_{i\gamma} > 0$ (β and γ are classes that do not appear in the other list), then the element in the β^{th} row and γ^{th} column of CM is increased by $\frac{Z_{i\gamma}}{(\mu * \nu)}$.

If μ or ν are equal to 0, then the classes that appear in both lists are taken instead. Figure 2 shows an example of how the CM is built based on the predictions of the BM and the noisy labels.

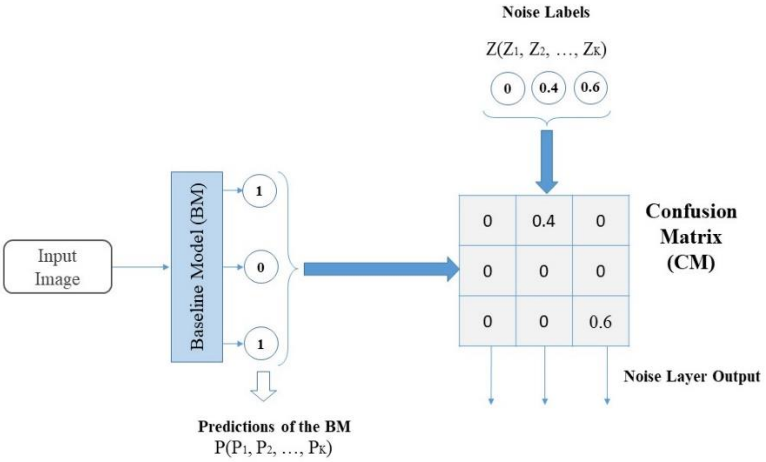

Fig. 2. A toy example of CM calculation. In such case, μ=1, since only one category (category 2) appears in Z but not in P; v=1 in a similar way.

3.2 Noise Layer

The CM is used here, since it is generally a simple approach to estimate the label noise and the relationships between the classes, with high accuracy. For each of the K outputs of the BM, a Dense layer with K units is added. For the kth output of the BM, the Dense layer that is appended is initialized with bias equal to the kth row of the CM. Each column, j, of the kth row of CM represents the probability that the kth output of the BM will go to the jth output of the Noise Layer as shown in Figure 2.

The FM, that contains the BM and the Noise Layer, is then retrained on the training set. It is worth noting here, that the weights of the BM that gave the best results were transferred to this FM. The Noise Layer is used to re-adjust the weights of the BM, by trying to estimate the noise distribution. Therefore, the predictions for the testing set are based on the output of the BM.

If the CM were diagonal, each BM output would only connect to the Noise Layer that depicts the same class. However, this is not the case. By connecting each BM output to several Noise Layer outputs, a correlation between these classes is also considered. If the element in the i^{th} row and j^{th} column of the CM is high, then the relation between the i^{th} BM output and the j^{th} Noise Layer is also high, meaning there is a chance that label i could possibly be mislabeled as class j.

4 Experimental Results

4.1 Dataset

The dataset we used contains in total 5778 RGB food images, which were taken under free living conditions by the end users of Oviva [38]. The database was annotated into

31 food categories by 5 inexperienced annotators and each image may contain more than one food category. For each food category in an image the mean of the annotations is taken. To quantitatively measure the noise level of the database, we randomly chose 200 food images from the database and conducted a consistency study among 5 annotators. According to the study, the Intersection over Union (IoU) of different annotators is around 0.8. We split the database into training and testing set, with 5485 and 293 food images, respectively. For the testing set, an additional experienced dietitian was involved to correct the annotations, so that the testing labels were much cleaner than those of the training set. Examples of images taken from the database are shown in Figure 3, along with the annotations.

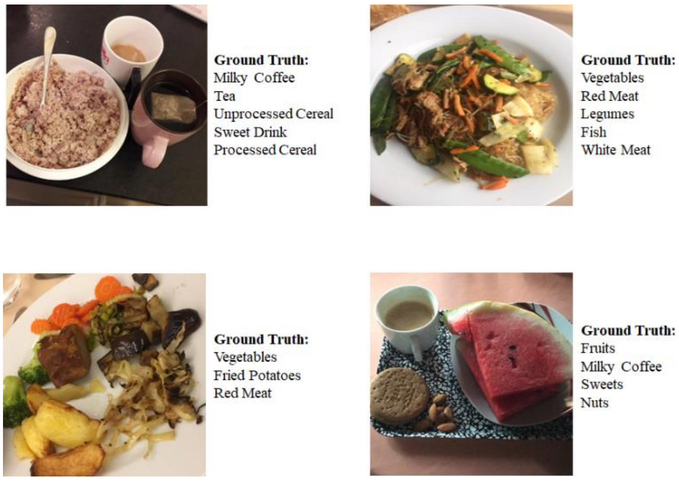

Fig. 3. Example images of the training set (upper row) and the testing set (lower row) of the database along with their annotations.

4.2 Evaluation metrics

The mean Average Precision (mAP) is typically used for the evaluation of multi-label classification tasks. mAP is calculated as described below:

$$mAP = \frac{1}{K}\sum_{k=1}^{K} mean\left(\max\left(P_{\mathrm{R}}^{k}\right)\right), \tag{1}$$

where K is the number of classes and $\max\left(P_{\mathrm{R}}^{k}\right)$ is the max precision for each recall value of category k.

The per-class Average Precision (AP) is also considered, which is, for each class k, the $mean\left(\max\left(P_{\mathrm{R}}^{k}\right)\right)$.

4.3 Results

We used the ResNet-101 and the InceptionV3 as BM, which were pretrained on the images of ImageNet. With the addition of a Dense layer with K=31 units, the BM were used to predict the probability of every image belonging to each class. For both BM, the Stochastic Gradient Descent was preferred as the optimizer, with learning rate set to 0.01, momentum 0.9 and decay 10^{-5}. The BM were supervised with a binary cross-entropy loss for 30 epochs and with a batch size equal to 8. The BM achieved a mAP of 0.466 on the testing set for the ResNet-101 and outperformed the InceptionV3, that achieved a mAP of 0.416.

After the BM was trained, it was used again to make predictions on the training set, in order to build the CM as suggested in Section 3.1. The noise layer was then appended to the BM. The same optimizer was used to train the FM for a batch size equal to 16 on 2 NVIDIA GeForce GTX TITAN X GPUs for another 30 epochs. The learning rate was set to 0.01 for the first 10 epochs and 0.005 for the rest. Changing the hyperparameters of the optimizer or the learning rate had minimal effect on the output of the model.

Table 1 shows a comparison between the two BM and their respective FM with the addition of the Noise Layer. The FM with the InceptionV3 as a BM reached its highest mAP of 0.499 at the 20th epoch and the FM with the ResNet-101 as a BM at the 10th epoch with mAP of 0.507. This is an 8.3% increase in mAP for the InceptionV3 model and a 4.1% increase for the ResNet-101 model.

Table 1. Comparison of mAP between the FM and the BM for the InceptionV3 and the ResNet-101 architecture.

Model	mAP
InceptionV3	0.416
InceptionV3 with Noise Layer	0.499
ResNet-101	0.466
ResNet-101 with Noise Layer	**0.507**

A comparison between the per-class AP of the BM and the FM for each category of the dataset is presented in Table 2, for both architectures. The foods are placed in order, so that the first category, “Vegetables”, has the most samples in the training set, while the last category has the fewest. In the 1st column of Table 2, the 31 food categories are shown, in the 2nd and 3rd column the AP for the BM and the FM are presented respectively and in the 4th and 5th column the number of samples that appear in the training and the testing sets. In the last column the subtraction of AP for the FM and the BM is calculated and it appears in white background if there is no difference between those two, in green if the AP is increased with the FM and in red if the AP is actually worse. In general, we observe that for the ResNet-101 there are 19 out of 31 food categories that have their AP increased after the addition of the Noise Layer. For the InceptionV3 there are 24 out of 31 food categories with their AP increased. Therefore, the FM can predict with higher consistency the most common food categories. Specifically, from the results, the FM can better distinguish between red meat and white meat compared to the BM and are able to predict with better accuracy the class yoghurt, which is considered a “difficult” category. Moreover, for both models, the AP is increased in most cases for the drinks, which are generally harder to recognize and often include label noise. Generally, in food categories that the annotators disagree the most tend to have their AP increased with the FM, meaning that the Noise Layer can effectively learn the noise distribution. For the food categories that annotators agree more, the results may vary, and, thus further investigation is needed.

Figure 4 shows some comparison examples of the testing set, regarding the results of the BM and the FM for the ResNet-101 architecture.

Table 2. The 31 food categories (1st column), the AP for each class for the BM and the FM (2nd and 3rd column), the samples of each class in the training and the testing set (4th and 5th column) and the difference between the AP for the FM and the BM, when using the ResNet-101 and the InceptionV3 architecture. Green color indicates an increase in performance by using FM, while red color means decrease in performance with FM.

Class	AP of BM		AP for FM		# of samples in the training set	# of samples in the testing set	Difference of AP between the FM and the BM	
	ResNet-101	InceptionV3	ResNet-101	InceptionV3			ResNet-101	InceptionV3
Vegetables	0.95	0.90	0.95	0.93	2505	140	0.00	0.03
Red meat	0.61	0.45	0.63	0.60	896	51	0.02	0.15
Sweets	0.61	0.55	0.57	0.66	863	36	-0.04	0.11
Yoghurt	0.33	0.45	0.40	0.55	832	22	0.07	0.10
Fruits	0.80	0.64	0.80	0.76	808	38	0.00	0.12
Cheese	0.67	0.62	0.72	0.59	707	40	0.05	-0.03
Non-white bread	0.74	0.64	0.64	0.71	652	31	-0.10	0.07
White meat	0.15	0.24	0.20	0.41	571	16	0.05	0.17
White bread	0.61	0.50	0.59	0.62	507	51	-0.02	0.12
Breaded Food	0.10	0.07	0.10	0.11	442	8	0.00	0.04
Milky coffee	0.26	0.24	0.27	0.23	378	7	0.01	-0.01
Legumes	0.17	0.23	0.22	0.35	375	9	0.05	0.12
Eggs	0.72	0.52	0.70	0.74	315	25	-0.02	0.22
Water	0.36	0.19	0.47	0.34	309	5	0.11	0.15
White pasta	0.57	0.58	0.60	0.58	308	21	0.03	0.00
Milk	0.43	0.42	0.47	0.51	279	10	0.04	0.09
Sweet drink	0.56	0.43	0.61	0.57	257	14	0.05	0.14
Non-fried potatoes	0.30	0.27	0.39	0.33	243	8	0.09	0.06
White rice	0.72	0.75	0.68	0.81	232	22	-0.04	0.06

Fish	0.19	0.20	0.26	0.21	224	15	0.07	0.01
Nuts	0.62	0.65	0.72	0.65	223	5	0.10	0.00
Unprocessed cereal	0.41	0.31	0.48	0.33	167	6	0.07	0.02
Non-white pasta	0.20	0.17	0.24	0.25	132	12	0.04	0.08
Fried potatoes	0.26	0.15	0.22	0.31	123	8	-0.04	0.16
Non-white rice	0.14	0.14	0.31	0.12	116	6	0.17	-0.02
Processed cereal	0.51	0.57	0.47	0.62	110	9	-0.04	0.05
Tea	0.46	0.47	0.50	0.67	96	6	0.04	0.20
Coffee	0.59	0.53	0.61	0.46	74	15	0.02	-0.07
Liquor	0.00	0.00	0.00	0.00	42	0	0.00	0.00
Wine	0.07	0.33	1.00	0.50	31	1	0.93	0.17
Beer	0.83	0.25	0.40	0.50	27	2	-0.43	0.25

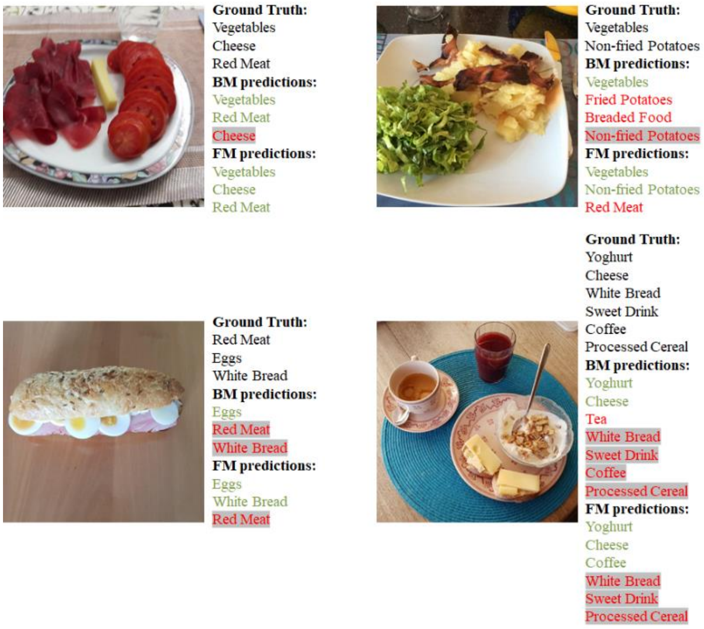

Fig. 4. Examples of images in the testing set along with predictions from the BM and the FM using the ResNet-101 architecture. The categories appear in green, red and red with grey background for correct, wrong and missing predictions, respectively.

5 Conclusions

In this paper we propose a method in order to deal with multi-label datasets containing label noise, where the noise distribution is unknown. We showed that by constructing a CM, the relations between the different classes, and, therefore, the existence of noise, can be observed. The FM, consisting of the BM and the Noise Layer yields an 8.3% increase in mAP compared to the BM when using the InceptionV3 architecture and a 4.1% increase when using ResNet-101. In addition to that, the Noise Layer was only used during the training phase, while it is not needed in the testing phase; thus, the proposed approach does not increase the computational time. In future, we intend to evaluate our proposed method on much bigger datasets.

Acknowledgments

This work was funded by Innosuisse under agreement n° 33780.1 IP-LS [www.innosuisse.ch] and it is a part of the medipiatto project [www.go-food.tech].

References

1. World Health Organization (WHO): https://www.who.int/, last accessed 2020/10/15
2. Centers for Disease Control and Prevention (CDC): https://www.cdc.gov/, last accessed 2020/10/15
3. Thompson, F. E., Subar, A. F.: Dietary assessment methodology. In *Nutrition in the Prevention and Treatment of Disease*, pp. 5-48, Academic Press (2017).
4. Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K.P.: Im2Calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1233-1241 (2015).
5. Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Ma, Y.: Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics*, pp. 37-48, Springer, Cham (2016).
6. Christodoulidis, S., Anthimopoulos, M., Mougiakakou, S.: Food recognition for dietary assessment using deep convolutional neural networks. In *International Conference on Image Analysis and Processing*, pp. 458-465, Springer, Cham (2015).
7. Dehais, J., Anthimopoulos, M., Shevchik, S., Mougiakakou, S.: Two-view 3D reconstruction for food volume estimation. In *IEEE transactions on multimedia*, 19(5), pp. 1090-1099 (2016).
8. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In *2012 IEEE International Conference on Multimedia and Expo*, pp. 25-30, IEEE (2012).
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), pp. 1261-1271 (2014).
10. Kawano, Y., Yanai, K.: Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, 74(14), pp. 5263-5287 (2015).

11. Lu, Y., Stathopoulou, T., Vasiloglou, M. F., Pinault, L. F., Kiley, C., Spanakis, E. K., Mougiakakou, S.: goFOODTM: An Artificial Intelligence System for Dietary Assessment. *Sensors*, 20(15), 4283 (2020).
12. Kagaya, H., Aizawa, K., Ogawa, M.: Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1085-1088 (2014).
13. Martinel, N., Foresti, G. L., Micheloni, C.: Wide-slice residual networks for food recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 567-576, IEEE (2018).
14. Anthimopoulos, M., Dehais, J., Diem, P., & Mougiakakou, S.: Segmentation and recognition of multi-food meal images for carbohydrate counting. In *13th IEEE International Conference on BioInformatics and BioEngineering*, pp. 1-4, IEEE (2013).
15. Aguilar, E., Remeseiro, B., Bolaños, M., Radeva, P.: Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia*, 20(12), pp. 3266-3275 (2018).
16. Bolaños, M., Ferrà, A., & Radeva, P.: Food ingredients recognition through multi-label learning. In *International Conference on Image Analysis and Processing*, pp. 394-402, Springer, Cham (2017).
17. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In *European conference on computer vision,* pp. 446-461, Springer, Cham (2014).
18. Martinel, N., Piciarelli, C., Micheloni, C., Luca Foresti, G.: A structured committee for food recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 92-100 (2015).
19. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9 (2015).
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778 (2016).
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z.: Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826 (2016).
22. Kawano, Y., & Yanai, K., Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. *In European Conference on Computer Vision,* pp. 3-17, Springer, Cham (2014).
23. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778-8788 (2018).
24. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. In: *IEEE transactions on neural networks and learning systems*, 25(5), 845-869 (2013).
25. Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596 (2014).
26. Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K.: Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5552-5560 (2018).
27. Northcutt, C. G., Jiang, L., Chuang, I. L.: Confident learning: Estimating uncertainty in dataset labels. arXiv preprint arXiv:1911.00068 (2019).

28. Jiang, L., Zhou, Z., Leung, T., Li, L. J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304-2313 (2018).
29. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. arXiv preprint arXiv:1803.09050 (2018).
30. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 839-847 (2017).
31. Lee, K. H., He, X., Zhang, L., & Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5447-5456 (2018).
32. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1431-1439 (2015).
33. Natarajan, N., Dhillon, I. S., Ravikumar, P. K., Tewari, A.: Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196-1204 (2013).
34. Joulin, A., Van Der Maaten, L., Jabri, A., Vasilache, N.: Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pp. 67-84, Springer, Cham (2016).
35. Sukhbaatar, S., & Fergus, R.: Learning from noisy labels with deep neural networks. arXiv preprint arXiv:1406.2080, 2(3), 4 (2014).
36. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer (2016).
37. Ciocca, G., Napoletano, P., & Schettini, R.: CNN-based features for retrieval and classification of food images. *Computer Vision and Image Understanding*, 176, pp. 70-77 (2018).
38. Oviva S.A., Zurich, Switzerland: https://oviva.com/global/