Development of a Food Image Recognition Algorithm Using Machine Learning

Serge Assaad (serge.assaad@duke.edu)
Duke University https://orcid.org/0000-0001-9310-0972
Lawrence Carin
Duke University
Anthony Joseph Viera
Duke Medicine

Research

Keywords: machine learning, diet tracking, nutrition apps

Posted Date: March 3rd, 2020

DOI: https://doi.org/10.21203/rs.3.rs-15859/v1

RESEARCH

Development of a Food Image Recognition Algorithm Using Machine Learning

Serge Assaad[1*], Lawrence Carin[1] and Anthony J Viera[2]

*Correspondence:
serge.assaad@duke.edu
[1]Department of Electrical & Computer Engineering, Duke University, Science Drive, 27708 Durham, NC, USA
Full list of author information is available at the end of the article

Abstract

Background: Researchers and consumers have limited options for objectively collecting or tracking data related to food choices.

Objective: To develop and pilot test an algorithm that could accurately categorize food items from a meal photograph.

Methods: We used a dataset of 7721 meal photographs taken by patrons in a cafeteria setting. We designed 22 broad categories recognizable by image that are parents of the original 1239 types of items in the photographs. We split the dataset into 3 mutually exclusive subsets: a training set (5250 images), a validation set (1312 images), and a test set (1159 images). Using a convolutional neural network and standard machine learning techniques, we tested the operating characteristics of the algorithm.

Results: Salad recognition had the lowest specificity (0.74), while multiple categories had specificities close to 1.0 (e.g. cereals, pastries, sushi, yogurt). Areas under the ROC curve (AUCs), reflecting trade-offs between sensitivity and specificity, ranged from 0.73 (for yogurt) to 0.97 (for sushi).

Conclusions: This work provides proof-of-concept for an algorithm that can categorize food items from a meal photograph.

Keywords: machine learning; diet tracking; nutrition apps

Introduction

Obesity and physical inactivity are leading causes of premature death in the United States and across the globe. The costs of obesity and its health related problems are also a substantial societal burden. Obese adults have more visits to physicians, and a higher number of inpatient hospital days per year [1],[2]. Given the public health and medical consequences of obesity and unhealthy eating, nutrition research is of critical importance.

Methods used to measure nutritional content of foods that people eat include various forms of food surveys. A food diary (or food/diet record) has people record all foods and drinks consumed over a specified time period (e.g. consecutive days). Measuring the items is critical to obtaining reliable estimates. The method does not rely on recall, but it is burdensome and the measurement process itself might alter eating behaviors. The Food Frequency Questionnaire [3] asks people to report how frequently they consumed certain foods and drinks over a specified time period. It is less burdensome and does not alter eating behaviors, but it relies on recall. Additionally, it is not as precise in terms of quantifying intake. The Automated Self-Administered 24-Hour (ASA24®) [4] Dietary Assessment Tool provides a more quantifiable estimate of intake while keeping burden fairly low. The ASA24 may capture food intake with less bias than food frequency questionnaires, but it still relies on recall.

A method that facilitates accurate capture of food information with little or no reliance on recall would be a valuable research tool. Such a method could also revolutionize our approach to helping people with their nutrition needs. An image-based dietary assessment tool could provide quantifiably accurate assessments with no reliance on recall and extremely low burden. We developed a prototype machine learning algorithm to demonstrate that foods can be recognized via a smartphone meal photograph, which can then be used to estimate calories and other nutritional

content. This paper reports the methods we used to employ machine learning to develop a prototype algorithm along with results of our initial testing.

Methods

Description of Image Dataset

Our image dataset was derived from a study [5] that used photographs of lunchtime meals to estimate calories purchased. The dataset contains 7721 images of meals, with each image containing several different food items. The labels for each image come from a list of food items from $F = 1239$ categories. Otherwise stated, we have a dataset $D = \{x_i, \tilde{y}_i\}_{i=1}^{N}$, where x_i is the i-th image in the dataset, and $\tilde{y}_i$ is a binary vector of length 1239 with ones at the indices corresponding to food items present in image $x_i \in \mathbb{R}^{224\times224\times3}$. Note that this could be framed as an object detection problem, but we do not have bounding boxes around the objects of interest (i.e. the food items), as that would require significant manual work to obtain. Instead, we were limited to binary labels indicating presence or absence of the food items. A few example pictures are shown in Figure 1.

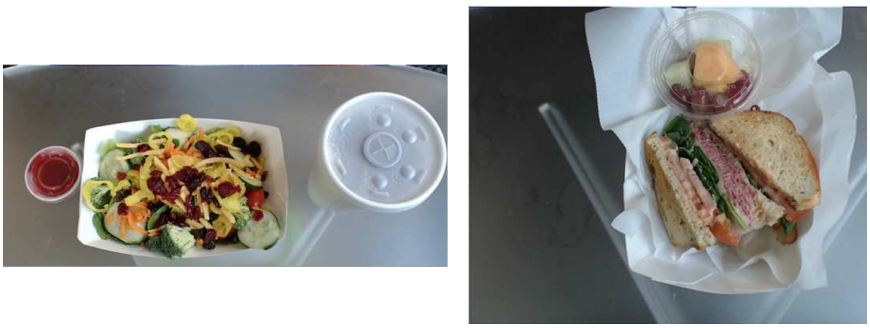

Figure 1 Example images from dataset.

Development of Broader Categories Recognizable by Image

Since there are $F = 1239$ categories for only $N = 7721$ images, we needed to make the categories coarser to have more examples per category to properly train a machine learning system. We designed $B = 22$ broader categories, which are parents

of the original $F = 1239$ categories. We achieved this task via a manually crafted assignment matrix $A \in \{0,1\}^{22\times 1239}$, which is essentially a binary table with a 1 at location (i, j) if original class j is a subcategory of coarse class i.

For example, the entry (Soup, 12oz Chicken Noodle) is a 1, but the entry (Desserts, Meat Lovers Pizza) is a 0 since "12oz Chicken Noodle" is a subcategory of "Soup", but "Meat Lovers Pizza" is not a subcategory of "Desserts".

In order to convert a label $\tilde{y}_i$ for an image x_i into a coarse label vector y_i, we defined the following:

$$y_i = \mathbb{1}(A\tilde{y}_i > 0) \tag{1}$$

where $\mathbb{1}(\cdot)$ is the elementwise indicator function. This process established a dataset of N images with corresponding coarse label vectors – i.e. $D = \{x_i, y_i\}_{i=1}^{N}$, with $x_i \in \mathbb{R}^{224\times 224\times 3}$ and $y_i \in \{0,1\}^{22}$.

Having reduced the number of categories from 1239 to 22, the number of instances of each label in the dataset is sufficient to train a deep learning system to label food items according to the broad categories (the number of instances for each coarse label are shown in Table 1).

Table 1 Sample counts for all 22 broad categories. Note the counts do not add up to $N = 7721$ since there are multiple food items per image.

Bread	Burger	Cereal	Cheese	Desserts	Dressing	Drink	Fried sides	Fruit	Meats	Pasta
926	318	56	901	551	2956	2715	1164	621	1825	400
Pastry	Pizza	Rice	Salad	Salty snacks	Sandwich	Soup	Sushi	Veggies	Wrap	Yogurt
204	224	856	4147	705	1671	899	163	2241	515	174

Data split & Augmentation

From our images and corresponding coarse labels, we split the dataset into 3 mutually exclusive subsets: a training set D_{train} (5250 images), a validation set D_{val} (1312 images), and a test set D_{test} (1159 images). Our system is trained on D_{train}. We use D_{val} to validate the performance of the system and tune hyperparameters. Finally, our results are reported on the test set D_{test}. Furthermore, we augment

our training set using random rotations of the images between 0 and 20 degrees, random horizontal & vertical flips (each with probability 50%), random horizontal shifts between $-0.2W$ and $0.2W$ pixels (where W is the image width), random vertical shifts between $-0.2H$ and $0.2H$ pixels (where H is the image height), and random central crops (between 1.1x and 1.6x zoom since most of the items are centrally located). These augmentations help the model gain a better understanding of the data distributions for different food items.

Model architecture

The convolutional neural network [6] architecture consists of the convolutional & pooling layers of MobileNetV2 [7] (since the intended use case would be on a mobile device), followed by an average pooling operation and a single fully connected layer – a diagram detailing the model architecture can be found in Figure 2. For

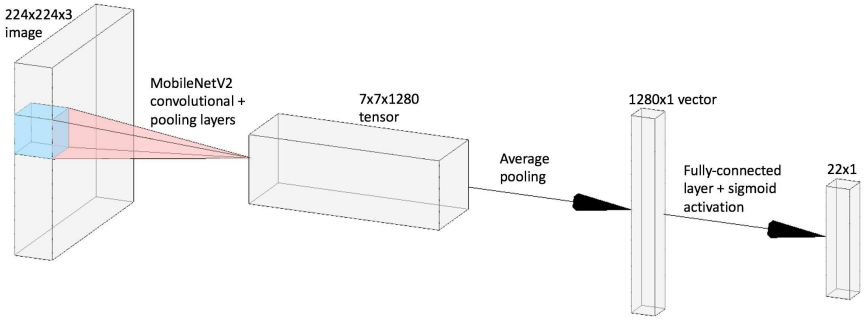

Figure 2 Model architecture

simplicity of notation, denote the convolutional & pooling layers of MobileNetV2 to be a function $MobileNet(\cdot) : \mathbb{R}^{224\times224\times3} \rightarrow \mathbb{R}^{7\times7\times1280}$. Specifically, the model is:

$$a = MobileNet(x) \tag{2}$$

$$a_{avg}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} a^{(i,j,k)} \qquad \forall\ k \in \{1, ..., 1280\} \tag{3}$$

$$\hat{y} = \sigma(a_{avg} V + b) \tag{4}$$

where $\sigma(\cdot)$ is the elementwise sigmoid function, $V \in \mathbb{R}^{1280 \times 22}$ is a learned weight matrix, $b \in \mathbb{R}^{22}$ is a learned bias vector, $x \in \mathbb{R}^{224 \times 224 \times 3}$ is the input image, and $\hat{y} \in \mathbb{R}^{22}$ is the prediction vector. $a^{(i,j,k)}$ denotes the (i, j, k)-th element of a, and $a_{avg}^{(k)}$ denotes the k-th element of a_{avg}. Henceforth, we will use the notation $\hat{y} = CNN(x)$, where $CNN(\cdot) : \mathbb{R}^{224 \times 224 \times 3} \to \mathbb{R}^{22}$ is the function described in equations (2)-(4).

Loss & Training

We use an elementwise binary cross-entropy loss to train the model:

$$\hat{y}_i = CNN(x_i) \tag{5}$$

$$\mathcal{L}_i = -\frac{1}{22} \sum_{k=1}^{22} y_i^{(k)} \cdot \log(\hat{y}_i^{(k)}) + (1 - y_i^{(k)}) \cdot \log(1 - \hat{y}_i^{(k)}) \tag{6}$$

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i \tag{7}$$

Here, $\mathcal{L}$ represents the loss on a batch of size n, and $_i$ is used to denote the i-th image in the batch. We train the entire network for 200 epochs using the Adam optimizer [8] with a learning rate of 0.001 and a batch size of 32.

Threshold selection

To convert the vector $\hat{y}_i \in \mathbb{R}^{22}$ into a binary prediction vector $\hat{y}_i^B \in \{0, 1\}^{22}$, we have to select a threshold for each dimension/category of $\hat{y}_i$. To do so, we sweep over a range of thresholds between 0 and 1, and – for each dimension/category –

we select the threshold which maximized balanced accuracy for that category on the validation dataset D_{val}.

Results

Table 2 Per-class results. The rows are #Train (the number of training examples), followed by AUC, accuracy, sensitivity, specificity.

	Bread	Burger	Cereal	Cheese	Desserts	Dressing	Drink	Fried sides	Fruit	Meats	Pasta
#Train	625	222	41	597	373	1989	1851	791	421	1271	272
AUC	0.93	0.86	0.84	0.86	0.82	0.83	0.94	0.97	0.88	0.82	0.78
Acc.	0.89	0.91	0.95	0.85	0.79	0.75	0.89	0.91	0.92	0.78	0.84
Sens.	0.87	0.56	0.75	0.69	0.69	0.77	0.87	0.92	0.67	0.68	0.56
Spec.	0.89	0.93	0.95	0.87	0.80	0.73	0.91	0.91	0.94	0.81	0.85

	Pastry	Pizza	Rice	Salad	Salty Snacks	Sandwich	Soup	Sushi	Veggies	Wrap	Yogurt
#Train	151	153	587	2807	489	1158	618	100	1523	350	116
AUC	0.83	0.93	0.94	0.88	0.96	0.94	0.96	0.97	0.86	0.95	0.73
Acc.	0.90	0.96	0.86	0.79	0.92	0.88	0.94	0.96	0.76	0.86	0.91
Sens.	0.52	0.82	0.88	0.83	0.91	0.83	0.88	0.89	0.80	0.92	0.40
Spec.	0.90	0.96	0.85	0.74	0.92	0.90	0.95	0.96	0.75	0.85	0.92

We report our results on the test set D_{test}. The per-class Area Under the ROC Curve (AUC), accuracies, sensitivities, and specificities are shown in Table 2. Accuracy is a little misleading here since there are so many negative examples for each class (i.e. one could build a classifier which predicts 0 items for every image and still get good accuracy), so we instead look to the AUC values, as well as sensitivities and specificities to get a better sense of our model's performance.

AUC values ranged from 0.73 for yogurt to 0.97 for sushi. Figure 3 shows the ROC curves for each food category, which show that the model works very well for most items.

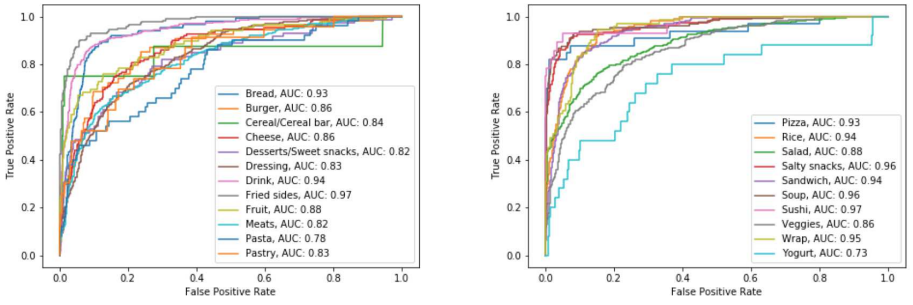

Figure 3 ROC curves for each class.

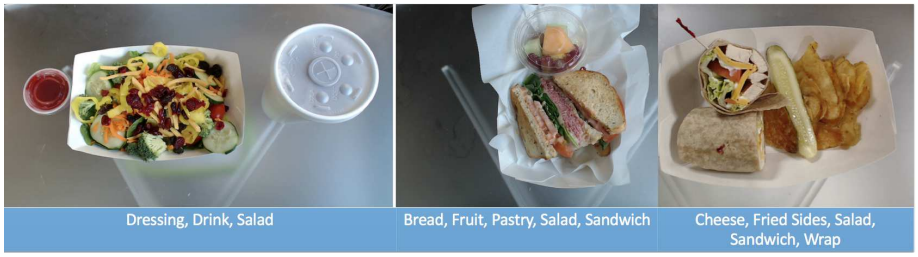

Figure 4 Example images with model predictions.

A few test set images and their corresponding predictions are shown in Figure 4. As can be seen, the algorithm correctly identifies the categories of foods in each photo. These categories are broad, however, and additional information would need to be gathered from the user to maximize the usefulness of the program.

Discussion

An algorithm that provides recognition and proper categorization of foods from a photograph holds great promise for research and practice applications. We developed such a prototype algorithm and demonstrated its ability to predict food categories with high accuracy. With the food categories linked to a database of nutritional content (and a portion-size estimator), the applications are numerous.

For the research community, the ability to capture food information without reliance on recall or burdensome diaries will be a valuable resource. Such a method will greatly mitigate measurement bias in such studies. From an end-user perspective, food recognition can be incorporated into programs and applications to track calories or other nutritional content, or provide guidance on specific diets to promote health (e.g., the DASH diet for hypertension) or weight loss.

The food categories are not granular enough alone to be useful for the kinds of applications we envision. However, programs could include a menu for users to input additional, or if needed, corrective, information. For example, the sandwich in Figure 4 is recognized and properly categorized as a sandwich, but the user would

have to indicate what is on the sandwich and the size of it. A program could be customizable to a person's usual preferences for food (e.g., the category "sandwich" could default to a "half-size ham & cheese on white bread", and "drink" could default to a "16 oz sweetened iced tea"). Additionally, crowd-sourcing could be employed to gather additional training data for the model.

Conclusion

We used machine learning to develop an algorithm that recognizes categories of food from a meal photograph. Our initial tests show a high degree of accuracy for placing food items in a meal photo into the correct categories. Next steps include incorporation of this type of algorithm into an application that research participants, and—when more refined—consumers and can utilize.

Declarations

Ethical Approval and Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of supporting data and materials

Data may be made available by contacting AJV.

Below are the details to access the code used for this project:

- Project name: Food Labeling with Deep Learning
- Project home page: https://github.com/sergeassaad/nutrition
- DOI: 10.5281/zenodo.3675840
- Operating system(s): Platform independent
- Programming language: Python
- Other requirements: Tensorflow, Keras
- License: MIT

Competing interests

The authors declare that they have no competing interests.

Funding

The PACE Study was funded by R01 CA184473-01A1 from the National Cancer Institute, National Institutes of Health. The funding body played no role in study design, data collection, analysis, interpretation of data, or writing the manuscript.

Author's contributions

AJV conceived of the study. LC, SA, and AJV designed the analysis plan. SA performed the analyses. SA and AJV drafted the manuscript. LC provided critical review of the manuscript.

Acknowledgements

Not applicable.

Author details

[1]Department of Electrical & Computer Engineering, Duke University, Science Drive, 27708 Durham, NC, USA. [2]Department of Family Medicine & Community Health, Duke University, 2200 West Main Street, Suite 400, 27705 Durham, NC, USA.

References

1. Quesenberry Jr, C.P., Caan, B., Jacobson, A.: Obesity, Health Services Use, and Health Care Costs Among Members of a Health Maintenance Organization. Archives of Internal Medicine **158**(5), 466–472 (1998). doi:10.1001/archinte.158.5.466
2. Thompson, D., Brown, J.B., Nichols, G.A., Elmer, P.J., Oster, G.: Body mass index and future healthcare costs: A retrospective cohort study. Obesity Research **9**(3), 210–218 (2001). doi:10.1038/oby.2001.23
3. Gonzalez Carrascosa, R., Garcia Segovia, P., Martinez Monzo, J.: Paper and pencil vs online self-administered food frequency questionnaire (FFQ) applied to university population: a pilot study. Nutricion hospitalaria **26**(6), 1378–1384 (2011). doi:10.1590/S0212-16112011000600027
4. Subar, A.F., Kirkpatrick, S.I., Mittl, B., Zimmerman, T.P., Thompson, F.E., Bingley, C., Willis, G., Islam, N.G., Baranowski, T., McNutt, S., Potischman, N.: The automated self-administered 24-hour dietary recall (ASA24): A resource for researchers, clinicians, and educators from the national cancer institute. Journal of the Academy of Nutrition and Dietetics **112**(8), 1134–1137 (2012). doi:10.1016/j.jand.2012.04.016
5. Viera, A.J., Tuttle, L., Olsson, E., Gras-Najjar, J., Gizlice, Z., Hales, D., Linnan, L., Lin, F.-C., Noar, S.M., Ammerman, A.: Effects of physical activity calorie expenditure (PACE) labeling: study design and baseline sample characteristics. BMC Public Health **17**(1) (2017). doi:10.1186/s12889-017-4710-0
6. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient-based learning. In: Shape, Contour and Grouping in Computer Vision, p. 319. Springer, London, UK, UK (1999). http://dl.acm.org/citation.cfm?id=646469.691875
7. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR **abs/1801.04381** (2018). 1801.04381
8. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2014). 1412.6980

Figure legend

- **Figure 1:** Example images from the dataset. The corresponding labels are (Dressing, Salad, Drink) for the left image, and (Sandwich, Fruit) for the right image.
- **Figure 2:** Model architecture used for image labeling. The model employs the convolutional and pooling layers of the MobileNetV2 architecture and an additional fully connected layer for prediction.
- **Figure 3:** Per-class ROC curves. Note, a perfect classifier is one for which the ROC curve reaches the top left corner (i.e. True Positive Rate=1.0, False Positive Rate=0.0), and for which AUC=1.0. The model shows strong performance for most food categories.
- **Figure 4:** Examples of images from the dataset with the corresponding predictions from the model.

Figures

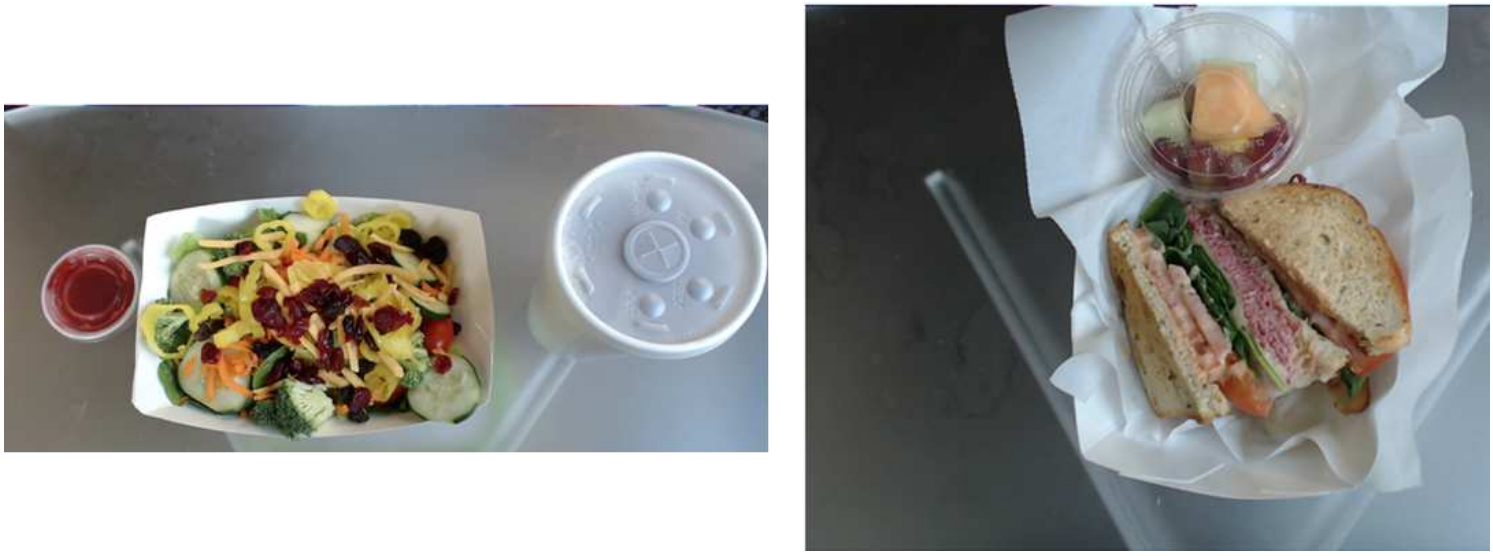

Figure 1

Example images from the dataset. The corresponding labels are (Dressing, Salad, Drink) for the left image, and (Sandwich, Fruit) for the right image.

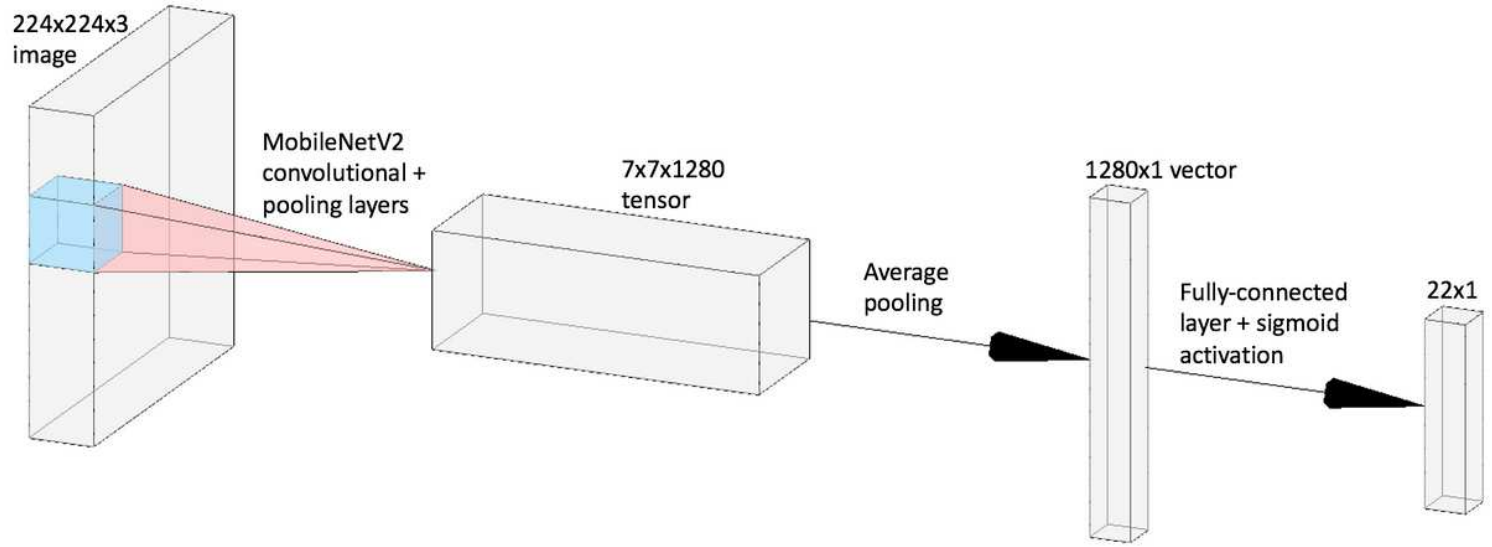

Figure 2

Model architecture used for image labeling. The model employs the convolutional and pooling layers of the MobileNetV2 architecture and an additional fully connected layer for prediction.

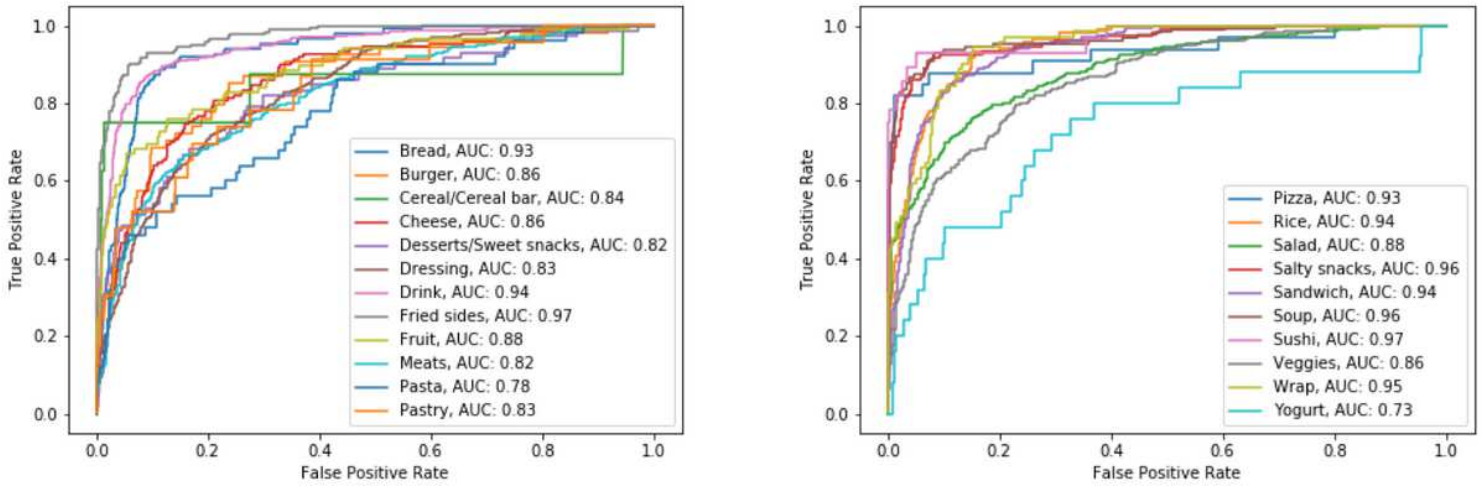

Figure 3

Per-class ROC curves. Note, a perfect classifier is one for which the ROC curve reaches the top left corner (i.e. True Positive Rate=1.0, False Positive Rate=0.0), and for which AUC=1.0. The model shows strong performance for most food categories.

Figure 4

Examples of images from the dataset with the corresponding predictions from the model.