Deep Learning based Ingredient Recognition and Calorie Estimation

Chrislin Priscilla, Shilpa Singh, Tanvi Patil

Abstract—Food safety and health is increasingly attracting attentions. People are shifting to vegan diet as vegans are less likely to develop heart disease, cancer, diabetes, and high blood pressure than meat-eaters. An effective computer vision method to recognize the dish and ingredients can efficiently help to evaluate to classify the dish as vegan or non-vegan and estimate the amount of calories. We propose DCNN-based deep architectures for simultaneous learning of ingredient recognition, food categorization, vegan non-vegan categorization, and calorie estimation. Our algorithm is evaluated using the VIREO Food-172 dataset which comprises of Chinese food images of highly complex dish appearance. This paper demonstrates the feasibility of ingredient recognition and sheds light on vegan non-vegan classification and calorie estimation. Our future idea is to improve the accuracy of the architectures and integrate our system into a real-world mobile application or web-based application to enhance, and improve the accuracy of current measurements of dietary intake in our practical daily lives.

1 Introduction

Diet is essential to a human's life. It is very vital for one to know what are they consuming and what is the calorie intake. With increasing awareness and consciousness on dietary intake, it becomes naturally important to employ deep learning techniques that can assure vegans that the food they are consuming has only animal-free ingredients. Ingredient recognition is to uncover the ingredients inside a dish (e.g., green pepper, black bean, crab) and is a much harder problem to solve than food recognition. While food recognition has been a common to scholars in this field, ingredient recognition has been shed less light. We propose in this paper that identifying ingredients will uncover more about the dish and provide more accurate classification of vegan dishes and calories estimation which is the novel idea in this paper. The solution can also help replace the traditional time-consuming and error prone food-log management systems which necessitates manual input of food-intake.

Ingredient recognition is challenged by the wildly different ways of mixing ingredients for the same food category. In other words, the mapping of Chinese food and categories is many to many. The size, shape, and other visual differences of ingredients makes it a task far harder than food categorization.

The paper experiments with single task learning architectures in learning the food, ingredients, and vegan non-vegan labels. We propose different ideas of shared and parallel DCNN and dense layers between food and ingredients and using the combination of output to estimate calories and classify vegan dishes. The VIREO Food-172 dataset includes around 66,000 Chinese food images in the training set, 11000 images in validation, and 34000 samples in test set. Dataset also provided 172 food categories and 353 ingredients.

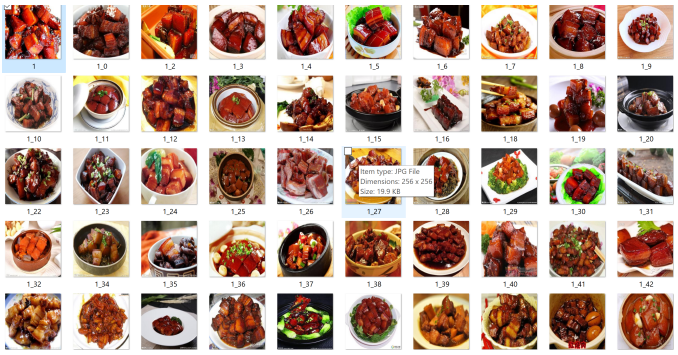

Fig. 1: VIREO Food-172 Dataset

We manually labelled ingredients as vegan or non-vegan. We programmed the true labels such that even if a single ingredient is non-vegan in a dish, we categorized the dish as non-vegan to ensure appropriate classification and mirror real world. Also, the true calories is calculated as the average of unit calories of all ingredients present in a dish and we solve a regression problem to estimate the calories per dish. Food categorization and ingredient detection is specifically a challenging problem as food categorization with minor ingredient differences becomes a complex prediction problem. If we consider different rice dishes, change in a single ingredient, changes the whole food category. In the figure given below we can see four some rice dishes belonging to different food categories.

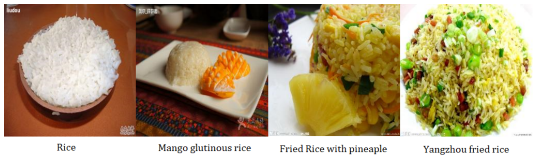

Fig. 2: Food categories consisting of rice

2 Background and Related Work

2.1 Multi Task Architecture in CNN

The initial work on ingredient recognition was by Jingjing Chen and Chong-Wah Ngo in [1] where they use stacked architectures on VIREO Food-172 dataset. The paper explains why ingredient recognition becomes vital to recognize recipes in a dish but visually different ingredients might be a challenge in getting good model accuracy. Hence, they couple food categorization problem, which is a single-label problem with ingredient recognition which is multi-label problem for simultaneous learning.
Similar approaches have been followed in [3], in creating late branching and early branching architectures for joint object categorization and pose estimation.

2.2 Food Image classification

Yuji Matsuda, et al., have used manifold ranking method in [4], considering co-occurrence statistics of food items for multiple food recognition. When the task is about classifying food images, Deep Convolution Neural Networks are proven to perform better than any other classification models as mentioned in [1]. In [2], Md Tohidul Islam, et al., have used pre-trained Inception V3 CNN model to classify food images from food-11 dataset, which is a more promising method. Whereas, Jingjing Chen and Chong-Wah Ngo, have used pre-trained VGG16 features coupled with different deep DCNN architectures for food and ingredient classification.

3 Architectures

3.1 Basic Approach

A higher-level schematic of our basic approach to food, ingredient, vegan-non vegan Categorization and calorie estimation is shown below.

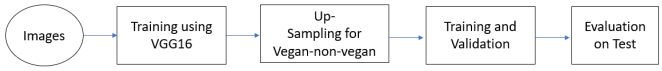

Fig. 3: Basic Approach

We first generate features using VGG16, which is then optionally up-sampled, to remove any biases in the vegan non-vegan dataset. This data is then used to train different models, and the results are validated through our validation dataset Finally, the results are tested using the test dataset provided by VIREO Food-172.

3.2 Features Generation

Deep learning systems and models are layered architectures that learn different features at different layers (hierarchical representations of layered features). These layers are then finally connected to a last layer to get the final output. This layered architecture allows us to utilize a pre-trained network such as VGG without its final layer as a fixed feature extractor for our problem specific task. We have used VGG pre-trained model as a feature extractor pre-processor where we removed the upper layers of VGG and used its lower layers to extract more generic features from our images. VGGNet consists of 16 convolution layers and is very appealing because of its very uniform architecture. It helped us to reduce training time by extracting the low level features which are more generic to image classification task before hand and use them as input to our domain-specific classification task.

3.3 Single Task Architecture

After training the images from VGG16, we created independent CNN architectures for food and ingredient categorization. The architecture contained 3-4 CNN layers with a softmax layer for food categorization and sigmoid layer for ingredients. We had to add additional CNN layers, as the input to our model were low level features extracted from VGG and we needed to fine tune these features for our problem specific task by adding more CNN layers. Sigmoid was chosen for ingredients, as it is a multi-label classification problem and there can be multiple ingredients present in a single food image.There were 353 unique ingredients in the entire dataset, so the output vector was 353 dimensional. Using sigmoid activation with binary crossentropy loss, we eventually turned this classification problem into 353 binary classification problems.

We did some initial analysis on the data to decide what would be the best evaluation metric to evaluate the performance of our models and see if there was any class imbalance issue in our dataset. We found that the food category was more or less evenly distribution across the data samples, but there was class imbalance issue with ingredients.There were around 11,000 samples belonging to one ingredient and 8000 samples belonging to other ingredient.The contribution of all the remaining ingredients were relatively less in the entire data samples.Figure 2 show the class distribution for food category and Figure 3 shows the class distribution for ingredient.

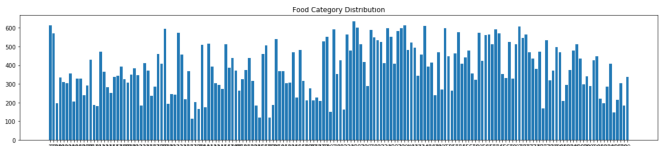

Fig. 4: Food Category Class Distribution

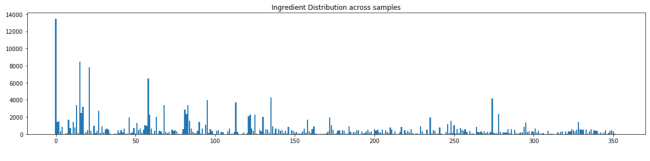

Fig. 5: Ingredient Class Distribution

Based on this analysis, we eventually came up with our evaluation metrics for food category and ingredient

recognition tasks. We chose top-5 accuracy to evaluate the performance of food category and F1-score to evaluate the performance of ingredient recognition task.

Our dataset was highly biased when we looked at Vegan vs Non-vegan data. The ratio of vegan to non vegan data was close to 1:5, due to which we were facing some performance issues, and the recall for Vegan category was 0. So, we augmented our dataset by adding 5 sets of randomly rotated images (an example is shown in the figure given below) for every vegan category image in our dataset. After completing this process we obtained around 120,000 images. As we were not able to load and process the whole data, we randomly sampled 80,000 observations, which had nearly balanced data. The initial macro average improved from 48 to 61 due to these changes, and recall improved to 21 from 0. Still, a low recall denotes that it is really difficult to differentiate Vegan vs Non-Vegan images.

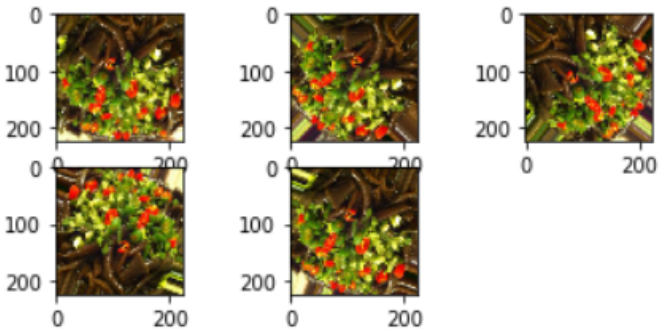

Fig. 6: Sample image with random rotations

3.4 Multi Task Architecture: Food and Ingredients

As proposed earlier, we built various CNN architectures combining food and ingredient recognition layers where predictions of each recognition system influence each other directly or indirectly.

For instance, the first design as shown in figure 7 is a stacked architecture that presents a direct influence where the output of ingredient is fed as input to food in A1 and vice versa in A2. In A2, the input from VGG16 trained features are fed to three Conv2D followed by a dense layer without any activation. This layer was applied a softmax activation to provide resultant food category. Simultaneously, the dense layer without activation is also passed to few more dense layers before being applied to sigmoid activation to output the probabilities for different ingredients. The direct links in this type of model faces challenge as a single dish can have combination of different ingredients. With the sigmoid activation function at the ingredient layer, the models output the probability of each class as bernoulli distribution, perfectly suited for multi-label classification.

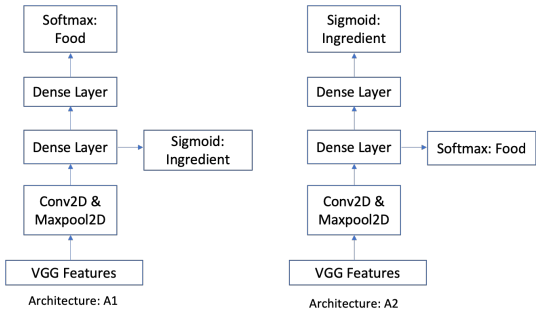

Fig. 7: Architecture A1 and A2

The next architecture, Fig 8, was designed with the idea that both food and category share all the convolution and dense layers. The final shared dense layer is then fed to a dense layer with sigmoid for ingredients and a dense layer with softmax for food category.This is called hard-parameter sharing.This kind of architecture greatly reduces overfitting.

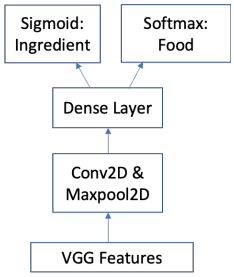

Fig. 8: Architecture B

Architecture C as shown in Fig 9 allows each of food and ingredients to have their own layers to train the parameters. The input is directly fed into parallel layers between food and category.This is called soft-parameter sharing.

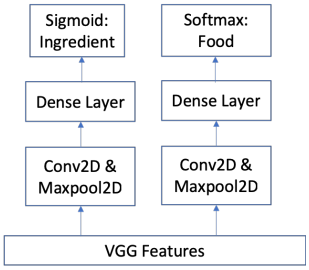

Fig. 9: Architecture C

Similar to Architecture C, Architecture D allows decoupling of intermediate layers. The intermediate layers are common to two recognition systems followed by private layers in end for each of them to learn the specialized features from the private layers.

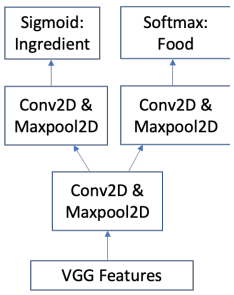

Fig. 10: Architecture D

3.5 Multi Task Architecture: Food, Ingredients, Vegan

Food and ingredient recognition architectures from above were modified to add layers for vegan non-vegan classification such that predictions of each recognition system influence each other directly or indirectly.

For the next three architectures, similar pattern from food and ingredient architectures were used where private layers of vegan were added accordingly.

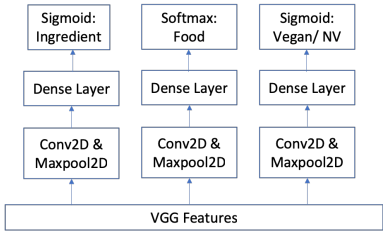

Fig. 11: Architecture C with Vegan Layers

3.6 Calorie Estimation

In order to estimate calories more accurately from the food category and ingredients, we tried different architectures with Multi-task learning as well as Single-task learning.The calorie value for 1 unit of each 353 ingredient was labeled using the common nutrition sites like myfitnesspal.com. These values were all numbers. Then we prepared the output Y to train the model by taking the average of the amount of calories of all the ingredients present in that dish. This was a normalization because for some of the ingredients the quantity could be much less than 1 unit, but it would not affect the model performance as the values will be normalized in validation as well as test set.

After that,we tried the architecture A and C for calorie estimation. In architecture A, the concatenated output from the dense layers of Food category and Ingredient Recognition after flattening and before performing the last layer activation was fed to the calorie estimation layer as an input. This was then connected to further dense layers and linear output activation. None of the multi-task architectures helped in the performance of the model and we got an MAE for calorie as high as 25.

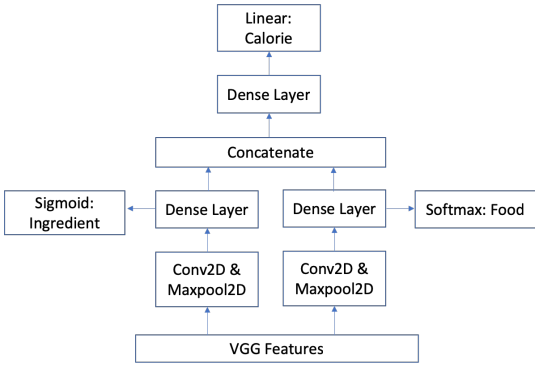

Fig. 12: Architecture A for Calorie Estimation

Since, calorie is an attribute of food category and ingredient, we tried another approach where we simply used the true labels of food category and ingredient vectors as input to a Dense layer with output as estimated calorie with linear activation.This architecture did not use any CNN but gave us good result with an MAE of 2.8. So, we can conclude that ingredients and food category true labels without images can be used to estimate the amount of calories of the food.

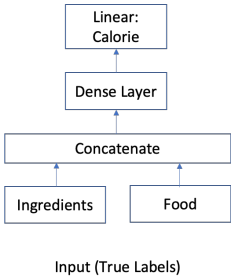

Fig. 13: Dense Architecture for Calorie without CNN

4 Loss

Cross-entropy loss, or log loss, is the standard method in measuring the performance of a classification model whose output is a probability value between 0 and 1.

For vegan, non-vegan classification model, the output is binary in nature with 1 for vegan and 0 for non-vegan. We used binary cross-entropy loss as shown in the equation below where y denotes the labels 1 and 0 and $P(y)$ denotes their probability.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

On the other hand, food categorization is a single-label problem. The 172 classes in our case in one hot encoded and categorical cross-entropy will compare the distribution of the predictions of each class with the true distribution. The class with the closest match is set to 1 and rest 0 for other classes. The equation of category cross-entropy is represented as follows:

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij}))$$

We again use binary cross-entropy for ingredient recognition which is a multi-label problem. The reason behind it is that the multi-label classifier can be considered as a multi separate binary classifier. For the 353 ingredient classes here, there will be 353 independent binary classifiers. When trained separately, probabilities for each label are generated. In this case, ingredients with labels having probabilities > 0.5 are considered the predicted ingredients for an image.

Finally, mean squared error (MSE) was chosen as the loss function for estimating calories in the regression model.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where y_i is the true value of calories which is average of calories/unit for every ingredient in the dish; $\hat{y}$ is the predicted value from the model.

5 Evaluation Metrics

F1 score: Harmonic mean of precision and recall. Used when true positives and false positives are crucial and when the classes are imbalanced. Hence, it is used as a metric to evaluate ingredients and vegan non-vegan.

Types of F1 Score:

- micro F1: computes f1 by taking account of total true positives, false negatives and false positives (no matter of the prediction for each label in the dataset)
- macro F1: computes f1 for each label, and returns the average without considering the proportion for each label in the dataset.
- weighted F1: weighted says the function to compute f1 for each label, and returns the average considering the proportion for each label in the dataset.

Top-5: Top 5 accuracy means that any of your model 5 highest probability answers must match the expected answer. Top 5 accuracy has been used as a metric in this paper to evaluate food classification.

Mean Absolute Error: It is the average over all the test samples of the absolute differences between true and predicted values where each sample has equal weight. We have used MAE as a metric to evaluate our regression model for estimating caloies.

6 Results

When we evaluated our models, we got the best results using architecture C which uses the early branching strategy. Using different architectures we have achieved better results as compared to our stand-alone models. As we can see here, the performance for all label classification improves as the architecture gets more complicated, except for model D.

TABLE 1: Results for Ingredient-Food architectures

Architecture	Ingredient F1-score	Food Top-5
Single	36.54	83.62
A1	45.99	77.32
A2	54.56	87.62
B	53.27	87.83
C	62.58	89.34
D	46.88	83.95

We selected a food category randomly from our test dataset, and analyzed results for ingredient categorization using Model B and C. Below are two sample cases for same food category 'Sauted Shrimp with Celery'. In the first figure, the ingredients are quite distinguishable so our model correctly identified the two ingredients present in image with a probability greater than 0.5.

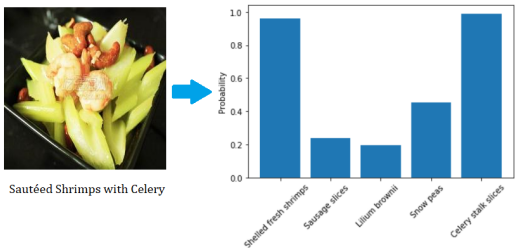

Fig. 14: Test example rightly classified for ingredient detection

However, when we look at the figure given below, we understand that it is really difficult to guess this dish right. It probably guesses celery as Minced Green onion and Shrimp as crushed Garlic.

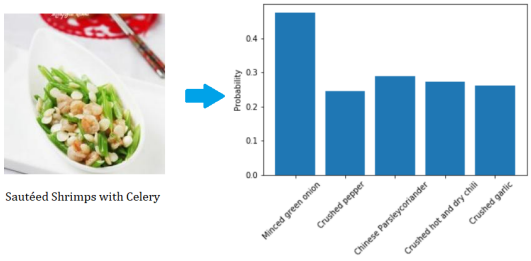

Fig. 15: Test example wrongly classified for ingredient detection

Adding Vegan categorization to existing architectures degraded the performance for food and ingredient categorization. Also, the performance of vegan, non-vegan classification was only marginally improved by the combined architectures.

TABLE 2: Results for Ingredient-Food-Vegan architectures

Architecture	Ingredient F1-score	Food Top-5	Vegan F1-score
Single	36.54	83.62	83.25
A1	43.27	73.03	84.90
A2	39.23	79.25	84.54
B	48.13	84.89	84.34
C	51.19	86.66	83.14
D	47.5	80.11	84.2

Adding Calorie Estimation to existing architectures degraded the performance for food and ingredient categorization. Also, it gave very high MAE for Calorie Estimation.When true labels of Ingredient Recognition and Food Categorization were used as features, we were able to obtain better results with a low MAE for calorie estimation.

TABLE 3: Results for Calorie Estimation

Architecture	Ingredient F1-score	Food Top-5	calorie MAE
Single with true label			2.85
A	38.6	78.5	25.7
C	48.23	87.5	28.17

7 Conclusion and Future Work

From the experiments, we ran, we can conclude that Multi-task learning can help to improve the performance of the model in some scenario when the features from one task can also be used for another task and are important representative of data. Probably, this can be the argument for the ingredient recognition to do well when coupled with food categorization in Architecture A1 where the output from the food category is fed as input to ingredient recognition. But the more important reason for the boost in performance can be attributed to the additional loss function which the model has to optimize. Additional loss function in principle does the same thing as regularization, it smooths the loss function, minimizes the complexity of model and gives informative prior to the model. This is the argument which can be given for the improvement in performance of ingredient recognition when combined with food categorization in architectures B,C and D.

Adding more tasks like vegan,non-vegan or calorie estimation to existing tasks are not helping to increase the performance of food category and ingredient recognition tasks, as the model tries to simultaneously learn a more general representation. When we train our model over an intersection of several tasks we push our learning algorithm to find a solution on a smaller area of representations on the intersection rather on a large area of a single task which could sometimes inhibit performance if the feature space is not too representative of all the tasks at hand. Also, we found that since amount of calories in a food item is an attribute of ingredients present in the food and food category, using them as direct features can help neural network to find the mapping function between them and calorie much easily.

In future, we will try to apply this multi-task approach in pairs by taking different combinations of input to see if we can gain any improvement in performance by forcing the model to find solution on a larger area of representation. Also, we can develop a mobile or a web-application for food log management which can get show calorie estimate and vegan non-vegan information by just taking food image as the input.

8 References

1) Jingjing Chen, Chong-Wah Ngo. 2016. Deep-based Ingredient Recognition for Cooking Recipe Retrieval
2) Md Tohidul Islam, B.M. Nafiz Karim Siddique, Sagidur Rahman, Taskeed Jabid. 2018. Food Image Classification with Convolutional Neural Network.
3) Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, Ahmed Elgammal. 2015. Convolutional Models for Joint Object Categorization and Pose Estimation
4) Yuji Matsuda and Keiji Yanai. 2012. Multiple-Food Recognition Considering Co-occurrence Employing Manifold Ranking
5) VireoFood-172 dataset [Online]. Available: http://vireo.cs.cityu.edu.hk/VireoFood172/
6) K. Aizawa and M. Ogawa. Foodlog: Multimedia tool for healthcare applications, 2015.
7) A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: towards an automated mobile vision food diary, 2015.
8) K. Yanai and Y. Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In Multimedia Expo Workshops (ICMEW), 2015.
9) H. Hoashi, T. Joutou, and K. Yanai. Image recognition of 85 food categories by feature fusion, 2010.
10) M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset, 2009.