pic2kcal: End-to-End Calorie Estimation From Food Pictures

Robin Ruede, Lukas Frank, Verena Heußer
Karlsruhe Institute of Technology

Abstract

Latest approaches to predicting calories of food usually use models that consist of several pipeline steps, such as segmenting the image, estimating the weight and classifying the ingredient. In this article we present a novel end-to-end approach to estimate the kcal directly from a picture. Since there is no large-scale publicly available dataset to train models on this task, we also collected data from recipes, including images, and matched the ingredients of the recipes with ground truth nutritional information of a food database.

1. Introduction

In recent years the awareness of healthier and more balanced diets has risen a lot. Tracking the exact amount and quality of food eaten is important for successfully following a diet, but doing this manually takes a lot of time and effort, leading to non-optimal results.

Currently, calorie tracking is mostly done by looking up specific ingredients and amounts. There are also a number of tools to help track calorie intake, such as the app Im2Calories by Google [1] from 2015, or CalorieMama [2], with varying automation.

In this work we propose a method to predict the caloric content of any prepared food based only on a single picture of one portion in an end-to-end fashion.

We collected a dataset of recipes, pictures, and nutritional properties. Then we experimented with various features and models to best predict calories and other nutritional information directly from an image. We measure our results objectively and show that multi-task learning improves the performance.

2. Dataset Extraction and Preprocessing

2.1. Collection

We collected a dataset from a popular German recipe website that contains ingredient lists, cooking instructions, and pictures of the resulting meals. The recipes are from many different cuisines and also include things like cakes, cocktails, and others. Most recipes have at least one picture. Smoe of the pictures are uploaded by the original author of the recipe or by third parties. Some of the pictures are of a single plate of food, others are for example of a whole casserole. We do not have any information about whether a picture contains a single portion. Around 10% of recipes contain a user-given value for how many calories per portion the recipe supposedly has.

2.2. Matching / Preprocessing

Since the dataset only has user-given calorie information for a small part of the data and doesn't include any details regarding the macronutrient composition, and since the user given information is often inaccurate (see Figure 1), we decided to match the list of ingredients against a database of nutritional values to sum up the proportions of macronutrients as well as the total calories.

Figure 1. A recipe with an obviously incorrect user-given calorie count.

To facilitate this, we collected a secondary dataset from a German website of nutritional values. The website contains values for the amount of fat, protein, and carbohydrates in

	per portion	per 100 g	per recipe
recipes count before	211k	211k	211k
removed no ings match	127k	127k	127k
removed no portions	31k	0k	0k
kcal mean	425 kcal	179 kcal	1791 kcal
kcal stddev	207 kcal	73 kcal	1007 kcal
kcal outliers	11k	14k	21k
final recipe count	42k	70k	63k
final data point count	179k	308k	267k

unmatched. If a recipe has at least one unmatched ingredient, it is discarded.

As a final step, we filter out all data points where the summed up calories of the recipe is outside of two standard deviations from the mean repeatedly until it converges. This is necessary because some recipes contain obviously wrong information (for example in a carrot cake recipe the author specified to use a million carrots).

2.3. Dataset Statistics

In total, the recipe website contains 330 thousand recipes. Of these, 210 thousand have at least one picture. Around 20 thousand recipes with pictures have user-given calorie information, though we didn't use these in the end. The recipes contain a total of 374 thousand unique ingredients. This high number is caused by slight differences in spelling or irrelevant details. In total, we collected 900 thousand pictures. On average, each recipe has 3 pictures.

The database of nutritional values contains a total of 390 thousand ingredients. Many of these are incomplete or duplicates, so we filter them by popularity to 123 thousand ingredients.

After matching the ingredients to the recipes, we have 50 to 85 thousand recipes with full nutritional information, depending on whether we aggregate calories per recipe, per portion or per 100g of raw mass (see Section 4). We lose 60% of recipes during matching because our matching discards recipes quickly when the ingredients don't fully match. This is so we can ensure we only retain data points that are accurate, and it could be improved with further tweaking. When aggregating per portion, we lose even more data points since we have to exclude all recipes where the user did not supply information about how many portions a recipe consists of.

In total, we have around 179 to 308 thousand data points (because each recipe has multiple images). We split these into train, validation and test set such that multiple pictures of the same recipe are in the same data split.

The 20 most common ingredients are shown in Table 2. Note how common baking ingredients are. This indicates a cake bias, i.e. the dataset may be biased towards sweet meals and desserts.

Count	Ingredient
119244	Salz
59066	Zucker
58185	Ei, vom Huhn
46069	Mehl
45891	Butter
41206	Zwiebel, frisch
24531	Milch (3,8 %)
24011	Vanillezucker
23476	Zucker
22822	Öl
22781	Paprika, orange
21348	Knoblauch
20359	Wasser
19935	Knoblauch, frisch
19336	Pfefferbreze
18928	Olivenöl
15966	Backpulver
15039	Sahne
14751	Zitrone, frisch
13077	Paprikapulver
12487	Gemüsebrühe, pflanzlich
12136	Backpulver
11960	Käse
11673	Kartoffeln
10926	Eigelb, vom Huhn
10780	Butter, Durchschnittswert
9591	Puderzucker
9439	Petersilie, frisch
8708	Zucchini, grün, frisch
8293	Mehl, Weizenmehl Typ 405

Table 2. Most common ingredients after matching.

3. Models

We followed an end-to-end approach to solve the calorie prediction problem of food images. To do so we used pretrained ResNet [6] and DenseNet [7] architectures as base models. We kept the feature extractor layers and replaced the last fully-connected classification layer. We tried solve the problem interpreting it on the one hand as a classification task and on the other hand as a regression problem. Furthermore we introduced additional learning feedback following a multi-task learning approac.

In the following, only the last layer of the neural network is described.

In the regression case we trained a model to predict the kcal information with one output neuron and additionally to predict protein, fat and carbohydrate information using three additional neurons. The two models were trained using a L1 and smooth L1 loss.

We adapted the base architecuters to the classification problem by quantizing the regression outputs. So we introduced 50 class buckets for each regression output. The models were trained using a cross entropy loss.

The multi-task model is based on the regression model including the nutritional information with additional binary outputs to predict the top n ingredients. The resulting layer has four regression outputs with 50 binary outputs. The used loss combines a smooth L1 loss for the regression outputs and a binary cross entropy loss for the top 50 ingredients. To get the same scaling of the two learning signals we scaled the binary cross entropy loss with a factor of 400.

$$\text{loss} = \text{L1} + 400 * \text{BCE}$$

As there are no reference papers working with similar approaches or similar data, the results could not be compared to other studies. Hence, a simple baseline was implemented to get evidence that our models actually learn and that they are better than random guessing.

The baseline for the kcal prediction basically is the mean of all samples in the train dataset. A basleine model would predict during inference only the mean of the already seen kcal values seen. The same baseline was used for predicting the nutritional data.

4. Experiments

We divided the generated dataset into train/test/validation (.7/.15/.15) splits. Our training set contains xxx samples with around xxx images for each recipe. The network was trained 40 epochs using a batch size of 50 samples each batch. The samples of the batches were shuffled every epoch and we evaluated the performance of the model every fiftieth batch. We implemented all networks using Pytorch.

To evaluate the performance of the model we trained several networks and run several experiments to evaluated them using firstly the evaluation data set to get quick feedback. To measure the performance of the model we only compared the given kcal information with the prediction of the network.

Firstly we used our raw data set to train the kcal-model. We wanted the network to predict the kcal information of the recipe visualized on the given input image. To perform well in this task the model needs to learn the concept of the recipe size and predict the calories according it. We assumed due to the amount of samples and the capacity of the model the problem is well learnable. Unfortunately the trained regression model performed not well on the task probably because of outlier recipes in our dataset with not valid kcal information provided by the users. Even after outlier removal, prediction of normalized kcal information of portion and trying a classification approach the model was only slightly better than a baseline model.

Method	kcal relative error
baseline	0.464
ours (kcal only)	0.361
ours (w/ macros)	0.352
ours (w/ macros+ings)	0.328

Table 3. Results per 100g. Note that multitask learning improves performance.

Second we evaluated if the additional nutritional information supports the networks capability to generalize on the recipe and portion size. Both the classification and regression objectives performed not well with the further information.

Lastly we reformulated the training objective to a slightly easier problem. We trained the network to predict the calory density of the visualized image. Because of the normalization the network only needs to grasp how many calories are in for instance 100g of the meal. This modification led to significant better results.

We could furthermore improve the results of the model using the multi-task approach. The top 50 ingredients of the recipes were injected as further information to support the model predicting the kcal information. We report the results of this model in the result section.

5. Results

For an objective comparison, we use the relative error of the kcal output neuron $\text{rel\_error} = 1 - \frac{\text{pred}}{\text{truth}}$.

Our results can be seen in Table 3. Example outputs can be seen in Figure 3.

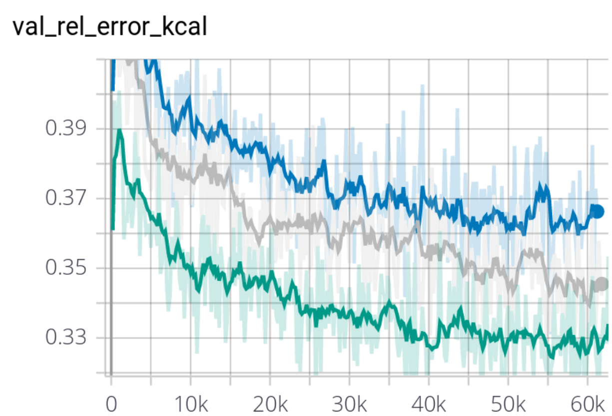

Figure 2. Relative validation error of the calorie prediction over training batches compared for a network predicting only calories (blue), predicting calories and macronutrients (gray), and predicting calories, macronutrients, and top100 ingredients (green). It can be seen that multi-task learning performs best.

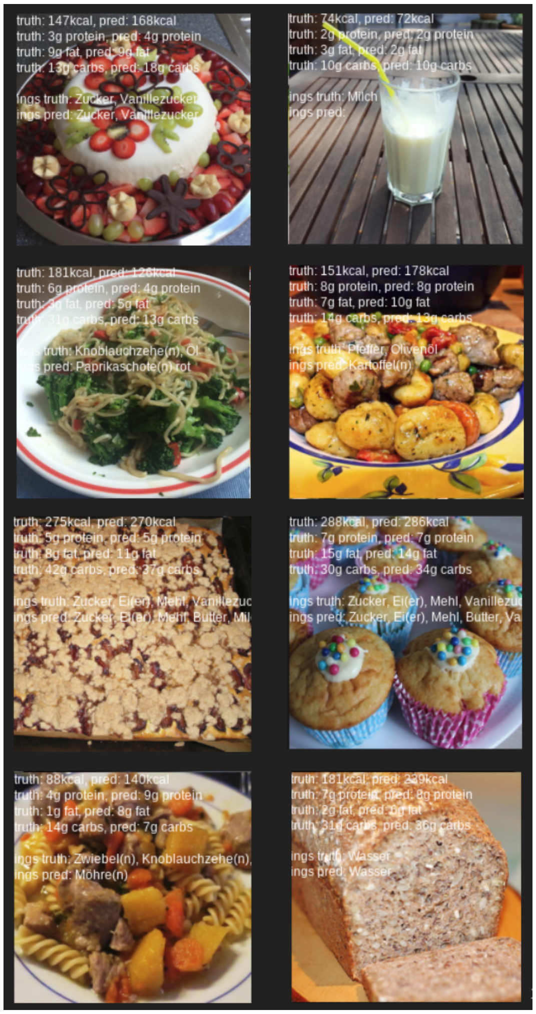

Figure 3. Some example results, showing predicted calories, fat, protein, carbohydrates and ingredients.

5.0.1 Different Architectures

- ResNet50, Resnet101, ResNext50p, Densnet121, Densent201
- per 100g vs per recipe vs per portion je mit Densnet121 und kcal+nut+ings
- predict only kcal, kcal+fat+protein+carbos, kcal+nut+ings je mit per 100g & Densenet121

6. Problems/Fails

Following we describe the most time consuming difficutlties we faced while working on the practical course task.

- **Scraping:** It was straight forward to crawl the recipes internet page because the graceful HTML structure. Whereas it was challenging to extract the nutritional data of the other website. The main problem was to find a proper method to extract the needed information out of the a HTML table which was modeld using several div tags.
- **Ingredient matching:** The matching of the recipe ingredients and the food database ingredients could not be solved with a simple method like Levenshtein distance. The obvious pairs could be matched but once for instance further information like *peeled potato* was provided often the optimal match was not found.
- **Kcal outlier:** It was not possible to use the raw user given kcal details because some of them were not accurate. The loss of the trained regression models exploded with bad results as consequence. Based on the matching we calculated kcal information and filtered the outlier recipes.
- **Tensorboard image visualization:** We used TensorBoard to be able to debug the models. Therefore we logged meta data including the images of the recipes. Because of an error related to the image normalization the images got destroyed.

7. Future Work

We currently only use a portion of the information in our extracted dataset. There are other interesting attributes such as the type of the meal (cake, side dish), the ingredient amounts, the cooking instructions, the rating, and further properties. These could be used to further improve our calorie prediction model as did adding prediction of the ingredients. The current dataset contains all available photos of each recipe. It may make sense to implement sanity checks to filter images out if they do not match the recipe in a proper way.

The cooking instructions could also be used to improve the ingredient matching by fine-tuning the text embedding model. We only used the pretrained Universal Sentence Encoder which is trained on online sources like Wikipedia, news, and discussion forums. Since these sources are very generic all ingredients are probably very close in the embedding space even though they may be different. The user-given free text formulation of the cooking instructions usually contain the names of the ingredients in text. Fine-tuning the Sentence Encoder with those instructions should help better encode similar ingredients (e.g. you usually add either baking powder or yeast at a specific cooking stage, so they are fairly similar ingredients).

Further problems related to food could also be approached using the dataset. For some people it may be interesting to know if a meal contains a specific ingredient because of allergies, if it is vegan or vegetarian, or if it fits a specific diet. The dataset provides needed information to train a variety of different models to solve problems related to food.

Currently our kcal prediction model is not highly optimized for the task since it is built on top of pretrained models. As we have shown it is beneficial to inject other data into the model therefore it may be interesting to do further investigation on different model architectures. For instance the representation of the top-n ingredient neurons could be changed from a binary value to a regression predicting the amount of the ingredient. It may also make sense to build entire new architectures using kernels with a size which match the requirements of predicting/classifying food images.

References

[1] A. Myers *et al.*, "Im2Calories: Towards an Automated Mobile Vision Food Diary," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1233–1241.

[2] *Calorie Mama Food AI - Food Image Recognition and Calorie Counter using Deep Learning*..

[3] *USDA Food Composition Databases*..

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[5] D. Cer *et al.*, "Universal Sentence Encoder," in *In submission to: EMNLP demonstration*, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[7] G. Huang, Z. Liu, L. v d Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.