Learning Cross-modal Embeddings for Cooking Recipes and Food Images

Amaia Salvador[1]* Nicholas Hynes[2]* Yusuf Aytar[2] Javier Marin[2] Ferda Ofli[3]
Ingmar Weber[3] Antonio Torralba[2]
[1]Universitat Politècnica de Catalunya [2]Massachusetts Institute of Technology
[3]Qatar Computing Research Institute, HBKU
amaia.salvador@upc.edu, nhynes@mit.edu, {yusuf,jmarin,torralba}@csail.mit.edu, {fofli,iweber}@qf.org.qa

Abstract

In this paper, we introduce Recipe1M, a new large-scale, structured corpus of over 1m cooking recipes and 800k food images. As the largest publicly available collection of recipe data, Recipe1M affords the ability to train high-capacity models on aligned, multi-modal data. Using these data, we train a neural network to find a joint embedding of recipes and images that yields impressive results on an image-recipe retrieval task. Additionally, we demonstrate that regularization via the addition of a high-level classification objective both improves retrieval performance to rival that of humans and enables semantic vector arithmetic. We postulate that these embeddings will provide a basis for further exploration of the Recipe1M dataset and food and cooking in general. Code, data and models are publicly available[1].

1. Introduction

There are few things so fundamental to the human experience as food. Its consumption is intricately linked to our health, our feelings and our culture. Even migrants starting a new life in a foreign country often hold on to their ethnic food longer than to their native language. Vital as it is to our lives, food also offers new perspectives on topical challenges in computer vision like finding representations that are robust to occlusion and deformation (as occur during ingredient processing).

The profusion of online recipe collections with user-submitted photos presents the possibility of training machines to automatically understand food preparation by jointly analyzing ingredient lists, cooking instructions and food images. Far beyond applications solely in the realm of culinary arts, such a tool may also be applied to the plethora of food images shared on social media to achieve insight into the significance of food and its preparation on public health [4] and cultural heritage [14]. Developing a tool for automated analysis requires large and well-curated datasets.

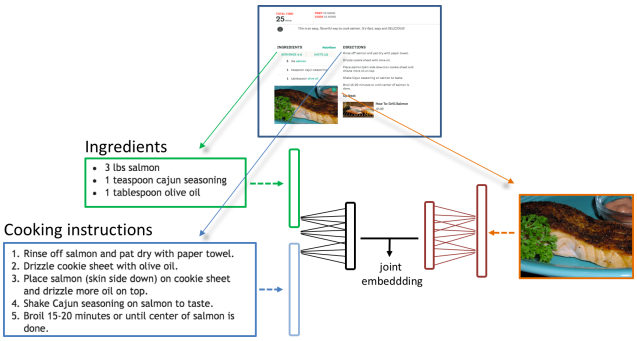


Figure 1: **Learning cross-modal embeddings** from recipe-image pairs collected from online resources. These enable us to achieve in-depth understanding of food from its ingredients to its preparation.

The emergence of massive labeled datasets [19, 26] and deeply-learned representations [10, 20, 5] have redefined the state-of-the-art in object recognition and scene classification. Moreover, the same techniques have enabled progress in new domains like dense labeling and image segmentation. Perhaps the introduction of a new large-scale food dataset–complete with its own intrinsic challenges–will yield a similar advancement of the field. For instance, categorizing an ingredient's state (e.g., sliced, diced, raw, baked, grilled, or boiled) provides a unique challenge in attribute recognition–one that is not well posed by existing datasets. Furthermore, the free-form nature of food suggests a departure from the concrete task of classification in favor of a more nuanced objective that integrates variation in a recipe's structure.

Existing work, however, has focused largely on the use of medium-scale datasets for performing categorization [1, 8, 16, 13]. For instance, Bossard et al. [1] introduced the Food-101 visual classification dataset and set a baseline of 50.8% accuracy. Even with the impetus for food image categorization, subsequent work by [13], [16] and [17] could only improve this result to 77.4%, 79% and 80.9%, respec-

*contributed equally.
[1]http://im2recipe.csail.mit.edu

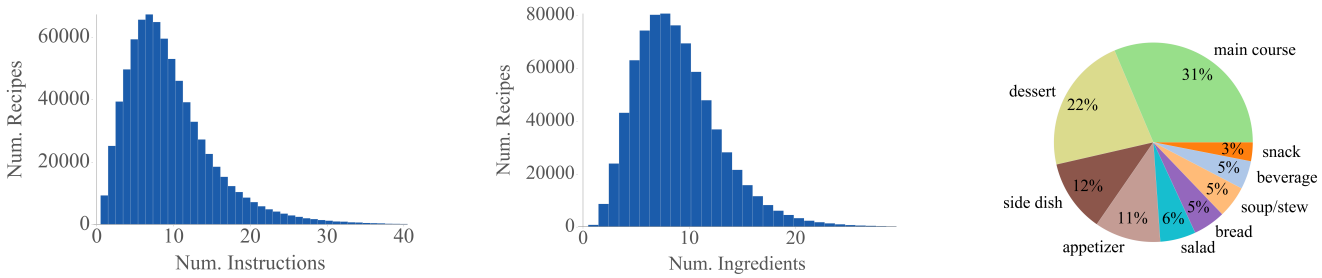


Figure 2: **Dataset statistics.** Prevalence of course categories and number of instructions and ingredients per recipe.

Partition	# Recipes	# Images
Training	720,639	619,508
Validation	155,036	133,860
Test	154,045	134,338
Total	1,029,720	887,706

Table 1: **Recipe1M dataset.** Number of samples in training, validation and test sets.

tively, which indicates that the size of the dataset may be the limiting factor. Although Myers et al. [16] build upon Food-101 to tackle the novel challenge of estimating a meal's energy content, the segmentation and depth information used in their work are not made available for further exploration.

In this work, we address data limitations by introducing the large-scale Recipe1M dataset which contains one million structured cooking recipes and their images. Additionally, to demonstrate its utility, we present the im2recipe retrieval task which leverages the full dataset–images and text–to solve the practical and socially relevant problem of demystifying the creation of a dish that can be seen but not necessarily described. To this end, we have developed a multi-modal neural model which jointly learns to embed images and recipes in a common space which is semantically regularized by the addition of a high-level classification task. The performance of the resulting embeddings is thoroughly evaluated against baselines and humans, showing remarkable improvement over the former while faring comparably to the latter. With the release of Recipe1M, we hope to spur advancement on not only the im2recipe task but also heretofore unimagined objectives which require a deep understanding of the domain and its modalities.

2. Dataset

Given the relevance of understanding recipes, it is surprising that there is not a larger body of work on the topic. We estimate that this is due to the absence of a large, general collection of recipe data. To our knowledge, virtually all of the readily available food-related datasets either contain only categorized images [16, 1, 8, 24] or simply recipe text [11]. Only recently have a few datasets been released that include both recipes and images. The first of which [23] has 101k images divided equally among 101 categories; the recipes for each are however raw HTML. In a later work, Chen and Ngo [6] present a dataset containing 110,241 images annotated with 353 ingredient labels and 65,284 recipes, each with a brief introduction, ingredient list, and preparation instructions. Of note is that the dataset only contains recipes for Chinese cuisine.

Although the aforementioned datasets constitute a large step towards learning richer recipe representations, they are still limited in either generality or size. As the ability to learn effective representations is largely a function of the quantity and quality of the available data, we create and release publicly a new, large-scale corpus of structured recipe data that includes over 1m recipes and 800k images. In comparison to the current largest dataset in this domain, Recipe1M includes twice as many recipes as [11] and eight times as many images as [6]. In the following subsections we outline how the dataset was collected and organized and provide an analysis of its contents.

2.1. Data Collection

The recipes were scraped from over two dozen popular cooking websites and processed through a pipeline that extracted relevant text from the raw HTML, downloaded linked images, and assembled the data into a compact JSON schema in which each datum was uniquely identified. As part of the extraction process, excessive whitespace, HTML entities, and non-ASCII characters were removed from the recipe text.

2.2. Data Structure

The contents of the Recipe1M dataset may logically be grouped into two layers. The first contains basic information including title, a list of ingredients, and a sequence of instructions for preparing the dish; all of these data are provided as free text. The second layer builds upon the first and includes any images with which the recipe is associated–these are provided as RGB in JPEG format. Additionally, a subset of

recipes are annotated with course labels (e.g., appetizer, side dish, dessert), the prevalence of which are summarized in Figure 2.

2.3. Analysis

The average recipe in the dataset consists of nine ingredients which are transformed over the course of ten instructions. Approximately half of the recipes have images which, due to the nature of the data sources, depict the fully prepared dish. Recipe1M includes approximately 0.4% duplicate recipes and 2% duplicate images (different recipes may share same image). Excluding those 0.4% recipes, 20% of recipes have non-unique titles but symmetrically differ by a median of 16 ingredients. 0.2% of recipes share the same ingredients but are relatively simple (e.g., spaghetti, granola), having a median of six ingredients. Regarding our experiments, we carefully removed any exact duplicates or recipes sharing the same image in order to avoid overlapping between training and test subsets. As detailed in Table 1, around 70% of the data is labeled as training, and the remainder is split equally between the validation and test sets.

In Figure 2, one can easily observe that the distributions of data are heavy tailed. For instance, of the 16k unique ingredients that have been identified, only 4,000 account for 95% of occurrences. At the low end of instruction count–particularly those with one step–one will find the dreaded *Combine all ingredients*. At the other end are lengthy recipes and ingredient lists associated with recipes that include sub-recipes. A similar issue of outliers exists also for images: as several of the included recipe collections curate user-submitted images, popular recipes like chocolate chip cookies have orders of magnitude more images than the average. Notably, 25% of images are associated with 1% of recipes while half of all images belong to 10% of recipes; the size of the second layer in number of unique recipes is 333k.

3. Learning Embeddings

In this section we introduce our neural joint embedding model. Here we utilize the paired (recipe and image) data in order to learn a common embedding space as sketched in Figure 1. Next, we discuss recipe and image representations and then we introduce our neural joint embedding model that builds upon recipe and image representations.

3.1. Representation of recipes

There are two major components of a recipe: its ingredients and cooking instructions. We develop a suitable representation for each of these components.

Ingredients. Each recipe contains a set of ingredient text as shown in Figure 1. For each ingredient we learn an ingredient level word2vec [15] representation. In order to do so, the actual ingredient names are extracted from each ingredient text. For instance in "2 tbsp of olive oil" the *olive_oil* is extracted as the ingredient name and treated as a single word for word2vec computation. The initial ingredient name extraction task is solved by a bi-directional LSTM that performs logistic regression on each word in the ingredient text. Training is performed on a subset of our training set for which we have the annotation for actual ingredient names. Ingredient name extraction module works with 99.5% accuracy tested on a held-out set.

Cooking Instructions. Each recipe also has a list of cooking instructions. As the instructions are quite lengthy (averaging 208 words) a single LSTM is not well suited to their representation as gradients are diminished over the many time steps. Instead we propose a two-stage LSTM model which is designed to encode a sequence of sequences. First, each instruction/sentence is represented as a *skip-instructions* vector and then an LSTM is trained over the sequence of these vectors to obtain the representation of all instructions. The resulting fixed-length representation is fed into to our joint embedding model (see instructions-encoder in Figure 3).

Skip-instructions. Our cooking instruction representation, referred as *skip-instructions*, is the product of a sequence-to-sequence model [21]. Specifically, we build upon the technique of skip-thoughts [9] which encodes a sentence and uses that encoding as context when decoding/predicting the previous and next sentences. Our modifications to this method include adding start- and end-of-recipe "instructions" and using an LSTM instead of a GRU. In either case, the representation of a single instruction is the final output of the encoder. As before, this is used as the instructions input to our embedding model.

3.2. Representation of food images

For the image representation we adopt two major state-of-the-art deep convolutional networks, namely VGG-16 [20] and Resnet-50 [5] models. In particular, the deep residual networks have a proven record of success on a variety of benchmarks [5]. Although [20] suggests training very deep networks with small convolutional filters, deep residual networks take it to another level using ubiquitous identity mappings that enable training of much deeper architectures (e.g., with 50, 101, 152 layers) with better performance. We incorporate these models by removing the last softmax classification layer and connecting the rest to our joint embedding model as shown in the right side of Figure 3.

4. Joint Neural Embedding

Building upon the previously described recipe and image representations, we now introduce our joint embedding method. The recipe model, displayed in Figure 3, includes two encoders: one for ingredients and one for instructions,

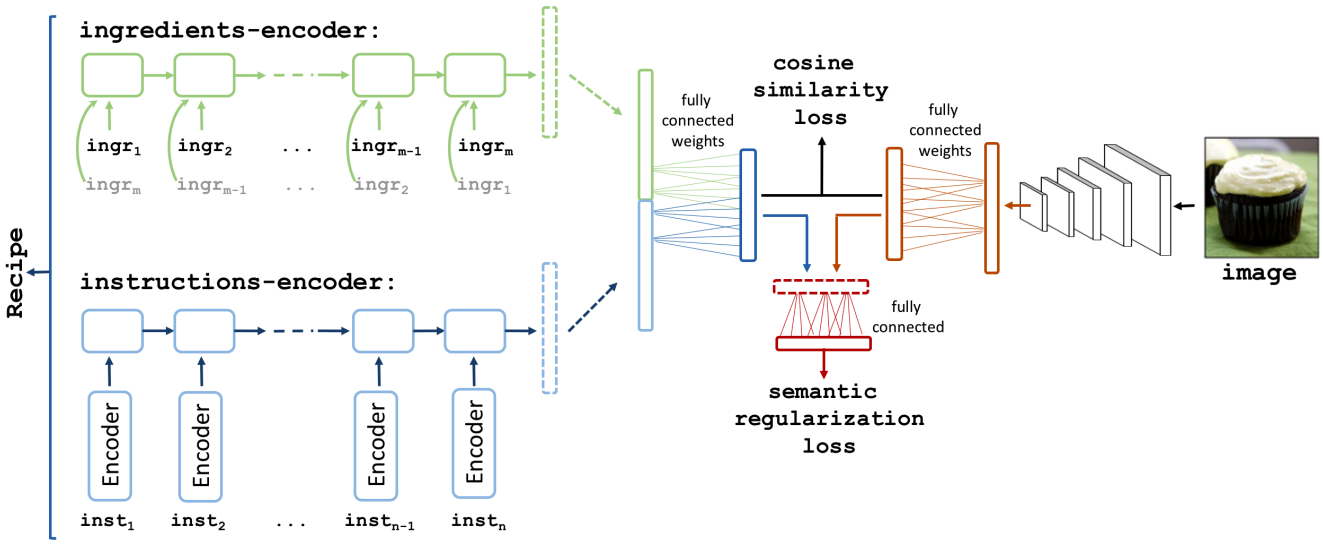


Figure 3: **Joint neural embedding model with semantic regularization.** Our model learns a joint embedding space for food images and cooking recipes.

the combination of which are designed to learn a recipe level representation. The ingredients encoder combines the sequence of ingredient word vectors. Since the ingredient list is an unordered set, we choose to utilize a bidirectional LSTM model, which considers both forward and backward orderings. The instructions encoder is implemented as a forward LSTM model over skip-instructions vectors. The outputs of both encoders are concatenated and embedded into a recipe-image joint space. The image representation is simply projected into this space through a linear transformation. The goal is to learn transformations to make the embeddings for a given recipe-image pair "close."

Formally, assume that we are given a set of the recipe-image pairs, (R_k, v_k) in which R_k is the k^{th} recipe and v_k is the associated image. Further, let $R_k = (\{s_k^t\}_{t=1}^{n_k}, \{g_k^t\}_{t=1}^{m_k}, v_k)$, where $\{s_k^t\}_{t=1}^{n_k}$ is the sequence of n_k cooking instructions, $\{g_k^t\}_{t=1}^{m_k}$ is the sequence of m_k ingredient tokens. The objective is to maximize the cosine similarity between positive recipe-image pairs, and minimize it between all non-matching recipe-image pairs, up to a specified margin.

The ingredients encoder is implemented using a bi-directional LSTM: at each time step it takes two ingredient-word2vec representations of g_k^t and g_k^{m-t+1}, and eventually it produces the fixed-length representation h_k^g for ingredients. The instructions encoder is implemented through a regular LSTM. At each time step it receives an instruction representation from the skip-instructions encoder, and finally it produces the fixed-length representation h_k^s. h_k^g and h_k^s are concatenated in order to obtain the recipe representation h_k^R. Then the recipe and image representations are mapped into the joint embedding space as: $\phi^R = W^R h_k^R + b^R$ and $\phi^v = W^v v_k + b^v$, respectively. W^R and W^v are embedding matrices which are also learned. Finally the complete model is trained end-to-end with positive and negative recipe-image pairs (ϕ^R, ϕ^v) using the cosine similarity loss with margin defined as follows:

$$L_{cos}((\phi^R, \phi^v), y) = \begin{cases} 1 - \cos(\phi^R, \phi^v), & \text{if } y = 1 \\ \max(0, \cos(\phi^R, \phi^v) - \alpha), & \text{if } y = -1 \end{cases}$$

where $\cos(.)$ is the normalized cosine similarity and α is the margin.

5. Semantic Regularization

We incorporate additional regularization on our embedding through solving the same high-level classification problem in multiple modalities with shared high-level weights. We refer to this method as semantic regularization. The key idea is that if high-level discriminative weights are shared, then both of the modalities (recipe and image embeddings) should utilize these weights in a similar way which brings another level of alignment based on discrimination. We optimize this objective together with our joint embedding loss. Essentially the model also learns to classify any image or recipe embedding into one of the food-related semantic categories. We limit the effect of semantic regularization as it is not the main problem that we aim to solve.

Semantic Categories. We start by assigning Food-101 categories to those recipes that contain them in their title. However, after this procedure we are only able to annotate 13% of our dataset, which we argue is not enough labeled data for a good regularization. Hence, we compose a larger set of semantic categories purely extracted from recipe titles. We first obtain the top 2,000 most frequent bigrams in recipe titles from our training set. We manually remove those that

	im2recipe				recipe2im			
	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10
random ranking	500	0.001	0.005	0.01	500	0.001	0.005	0.01
CCA w/ skip-thoughts + word2vec (GoogleNews) + image features	25.2	0.11	0.26	0.35	37.0	0.07	0.20	0.29
CCA w/ skip-instructions + ingredient word2vec + image features	15.7	0.14	0.32	0.43	24.8	0.09	0.24	0.35
joint emb. only	7.2	0.20	0.45	0.58	6.9	0.20	0.46	0.58
joint emb. + semantic	5.2	0.24	0.51	0.65	5.1	0.25	0.52	0.65

Table 2: **im2recipe retrieval comparisons.** Median ranks and recall rate at top K are reported for baselines and our method. Note that the joint neural embedding models consistently outperform all the baseline methods.

	Joint emb. methods	im2recipe			recipe2im		
		medR-1K	medR-5K	medR-10K	medR-1K	medR-5K	medR-10K
VGG-16	fixed vision	15.3	71.8	143.6	16.4	76.8	152.8
	finetuning (ft)	12.1	56.1	111.4	10.5	51.0	101.4
	ft + semantic reg.	8.2	36.4	72.4	7.3	33.4	64.9
ResNet-50	fixed vision	7.9	35.7	71.2	9.3	41.9	83.1
	finetuning (ft)	7.2	31.5	62.8	6.9	29.8	58.8
	ft + semantic reg.	5.2	21.2	41.9	5.1	20.2	39.2

Table 3: **Ablation studies.** Effect of the different model components to the median rank (the lower is better).

contain unwanted characters (e.g., *n'*, *!*, *?* or *&*) and those that do not have discriminative food properties (e.g., *best pizza*, *super easy* or *5 minutes*). We then assign each of the remaining bigrams as the semantic category to all recipes that include it in their title. By using bigrams and Food-101 categories together we obtain a total of 1,047 categories, which cover 50% of the dataset. *chicken salad*, *grilled vegetable*, *chocolate cake* and *fried fish* are some examples among the categories we collect using this procedure. All those recipes without a semantic category are assigned to an additional *background* class. Although there is some overlap in the generated categories, 73% of the recipes in our dataset (excluding those in the $background$ class) belong to a single category (i.e., only one of the generated classes appears in their title). For recipes where two or more categories appear in the title, the category with highest frequency rate in the dataset is chosen.

Classification. To incorporate semantic regularization to the joint embedding we use a single fully connected layer. Given the embeddings ϕ^v and ϕ^r, class probabilities are obtained with $p_r = W^c\phi^r$ and $p_v = W^c\phi^v$ followed by a softmax activation. W^c is the matrix of learned weights, which are shared between image and recipe embeddings to promote semantic alignment between them. Formally, we express the semantic regularization loss as $L_{reg}(\phi^r, \phi^v, c_r, c_v)$ where c_r,c_v are the semantic category labels for recipe and image, respectively. Note that c_r and c_v are the same if (ϕ^r, ϕ^v) is a positive pair. Then we can write the final objective as:

$$L(\phi^r, \phi^v, c_r, c_v, y) = L_{cos}((\phi^r, \phi^v), y) + \lambda L_{reg}(\phi^r, \phi^v, c_r, c_v)$$

Optimization. We follow a two-stage optimization procedure while learning the model. If we update both the recipe encoding and image network at the same time, optimization becomes oscillatory and even divergent. Previous work on cross-modality training [2] suggests training models for different modalities separately and fine tuning them jointly afterwards to allow alignment. Following this insight, we adopt a similar procedure when training our model. We first fix the weights of the image network, which are found from pre-training on the ImageNet object classification task, and learn the recipe encodings. This way the recipe network learns to align itself to the image representations and also learns semantic regularization parameters (W^c). Then we freeze the recipe encoding and semantic regularization weights, and learn the image network. This two-stage process is crucial for successful optimization of the objective function. After this initial alignment stage, we release all the weights to be learned. However, the results do not change much in this final, joint optimization.

Implementation Details All the neural models are implemented using the Torch7 framework[2]. The margin α is

[2] http://torch.ch/

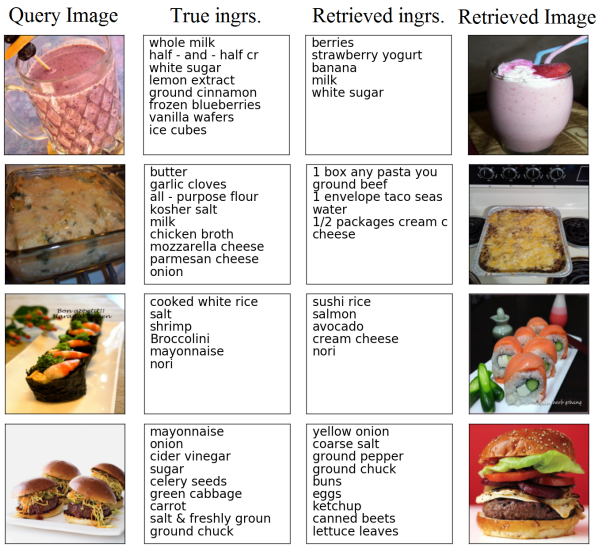


Figure 4: **Retrieval examples.** From left to right: (1) the query image, (2) its associated ingredient list, (3) the retrieved ingredients and (4) the image associated to the retrieved recipe.

selected as 0.1 in joint neural embedding models. The regularization hyperparameter is set as $\lambda = 0.02$ in all our experiments. While optimizing the cosine loss we pick a positive recipe-image pairs with 20% probability and a random negative recipe-image pair with 80% probability from the training set. The models are trained on 4 NVIDIA Titan X with 12GB of memory for three days.

6. Experiments

We begin with the evaluation of our learned embeddings for the im2recipe retrieval task. We then study the effect of each component of our model and compare our final system against human performance. We also analyze the properties of our learned embeddings through unit visualizations and vector arithmetics in the embedding space.

6.1. im2recipe retrieval

We evaluate all the recipe representations for im2recipe retrieval. Given a food image, the task is to retrieve its recipe from a collection of test recipes. We also perform recipe2im retrieval using the same setting. All results are reported for the test set.

Comparison with the baselines. Canonical Correlation Analysis (CCA) is one of the strongest statistical models for learning joint embeddings for different feature spaces when paired data are provided. We use CCA over many high-level recipe and image representations as our baseline. These CCA embeddings are learned using recipe-image pairs from the training data. In each recipe, the ingredients are represented with the mean word2vec across all its ingredients in the manner of [12]. The cooking instructions are represented with mean skip-thoughts vectors [9] across the cooking instructions. A recipe is then represented as concatenation of these two features. We also evaluate CCA over mean ingredient word2vec and skip-instructions features as another baseline. The image features utilized in the CCA baselines are the ResNet-50 features before the softmax layer. Although they are learned for visual object categorization tasks on ImageNet dataset, these features are widely adopted by the computer vision community, and they have been shown to generalize well to different visual recognition tasks [3].

For evaluation, given a test query image, we use cosine similarity in the common space for ranking the relevant recipes and perform im2recipe retrieval. The recipe2im retrieval setting is evaluated likewise. We adopt the test procedure from image2caption retrieval task [7, 22]. We report results on a subset of randomly selected 1,000 recipe-image pairs from the test set. We repeat the experiments 10 times and report the mean results. We report median rank (MedR), and recall rate at top K (R@K) for all the retrieval experiments. To clarify, R@5 in the im2recipe task represents the percentage of all the image queries where the corresponding recipe is retrieved in the top 5, hence higher is better. The quantitative results for im2recipe retrieval are shown in Table 2.

Our model greatly outperforms the CCA baselines in all measures. As expected, CCA over ingredient word2vec and skip-instructions perform better than CCA over word2vec trained on GoogleNews [15] and skip-thoughts vectors that are learned over a large-scale book corpus [9]. In 65% of all evaluated queries, our method can retrieve the correct recipe given a food image. The semantic regularization notably improves the quality of our embedding for im2recipe task which is quantified with the medR drop from 7.2 to 5.2 in Table 2. The results for recipe2im task are also similar to those in the im2recipe retrieval setting. Figure 4 compares the ingredients from the original recipes (true recipes) with the retrieved recipes (coupled with their corresponding image) for different image queries. As can be observed in Figure 4, our embeddings generalize well and allow overall satisfactory recipe retrieval results. However, at the ingredient level, one can find that in some cases our model retrieves recipes with missing ingredients. This usually occurs due to the lack of fine-grained features (e.g., confusion between *shrimps* and *salmon*) or simply because the ingredients are not visible in the query image (e.g., *blueberries* in a *smoothie* or *beef* in a *lasagna*).

Ablation studies. We also analyze the effect of each component in our our model in several optimization stages. The results are reported in Table 3. Note that here we also report medR with $1K$, $5K$ and $10K$ random selections to show how the results scale in larger retrieval problems. As expected,

Figure 5: **Localized unit activations.** We find that ingredient detectors emerge in different units in our embeddings, which are aligned across modalities (e.g., unit 352: "cream", unit 22: "sponge cake" or unit 571: "steak").

visual features from the ResNet-50 model show a substantial improvement in retrieval performance when compared to VGG-16 features. Even with "fixed vision" networks the joint embedding achieved 7.9 medR using ResNet-50 architecture (see Table 3). Further "finetuning" of vision networks slightly improves the results. Although it becomes a lot harder to decrease the medR in small numbers, additional "semantic regularization" improves the medR in both cases.

6.2. Comparison with human performance

In order to better assess the quality of our embeddings we also evaluate the performance of humans on the im2recipe task. The experiments are performed through Amazon Mechanical Turk (AMT) service[3]. For quality purposes, we require each AMT worker to have at least 97% approval rate and have performed at least 500 tasks before our experiment. In a single evaluation batch, we first randomly choose 10 recipes and their corresponding images. We then ask an AMT worker to choose the correct recipe, out of the 10 provided recipes, for the given food image. This multiple choice selection task is performed 10 times for each food image in the batch. The accuracy of an evaluation batch is defined as the percentage of image queries correctly assigned to their corresponding recipe.

The evaluations are performed for three levels of difficulty. The batches (of 10 recipes) are randomly chosen from either all the test recipes (easy), recipes sharing the same course (e.g., soup, salad, or beverage; medium), or recipes sharing the name of the dish (e.g., salmon, pizza, or ravioli; hard). As expected–for our model as well as the AMT workers–the accuracies decrease as tasks become more specific. In both coarse and fine-grained tests, our method performs comparably to or better than the AMT workers. As hypothesized, semantic regularization further improves the results (see Table 4).

In the "all recipes" condition, 25 random evaluation batches (25×10 individual tasks in total) are selected from the entire test set. Joint embedding with semantic regularization performs the best with 3.2 percentage points improvement over average human accuracy. For the course-specific tests, 5 batches are randomly selected within each given meal course. Although, on average, our joint embedding's performance is slightly lower than the humans', with semantic regularization our joint embedding surpasses humans' performance by 6.8 percentage points. In dish-specific tests, five random batches are selected if they have the dish name (e.g., pizza) in their title. With slightly lower accuracies in general, dish-specific results also show similar behavior. Particularly for the "beverage" and "smoothie" results, human performance is better than our method, possibly because detailed analysis is needed to elicit the homogenized ingredients in drinks. Similar behavior is also observed for the "sushi" results where fine-grained features of the sushi roll's center are crucial to identify the correct sushi recipe.

6.3. Analysis of the learned embedding

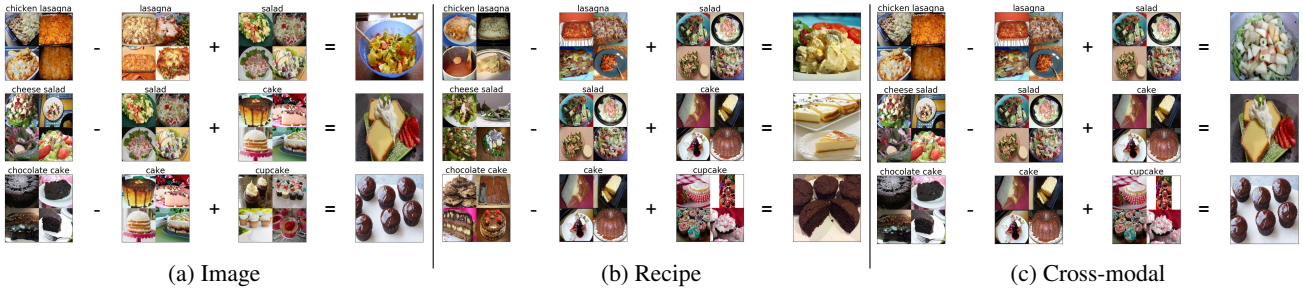
To gain further insight into our neural embedding, we perform a series of qualitative analysis experiments. We explore whether any semantic concepts emerge in the neuron activations and whether the embedding space has certain arithmetic properties.

Neuron Visualizations. Through neural activation visualization we investigate if any semantic concepts emerge in the

[3] http://mturk.com

	all recipes	course-specific recipes						dish-specific recipes								
		dessert	salad	bread	beverage	soup-stew	course-mean	pasta	pizza	steak	salmon	smoothie	hamburger	ravioli	sushi	dish-mean
human	**81.6** $\pm$ **8.9**	52.0	70.0	34.0	58.0	56.0	**54.0** $\pm$ **13.0**	54.0	48.0	58.0	52.0	48.0	46.0	54.0	58.0	**52.2** $\pm$ **04.6**
joint-emb. only	**83.6** $\pm$ **3.0**	76.0	68.0	38.0	24.0	62.0	**53.6** $\pm$ **21.8**	58.0	58.0	58.0	64.0	38.0	58.0	62.0	42.0	**54.8** $\pm$ **09.4**
joint-emb.+semantic	**84.8** $\pm$ **2.7**	74.0	82.0	56.0	30.0	62.0	**60.8** $\pm$ **20.0**	52.0	60.0	62.0	68.0	42.0	68.0	62.0	44.0	**57.2** $\pm$ **10.1**

Table 4: **Comparison with human performance on im2recipe task.** The mean results are highlighted as bold for better visualization. Note that on average our method with semantic regularization performs better than average AMT worker.



(a) Image (b) Recipe (c) Cross-modal

Figure 6: **Arithmetics** using image embeddings (left), recipe embeddings (middle) and cross-modal arithmetics between image and recipe embeddings (right). We represent the average vector of a query with the images from its 4 nearest neighbors. In the case of the arithmetic result, we show the nearest neighbor only.

neurons in our embedding vector despite not being explicitly trained for that purpose. We pick the top activating images, ingredient lists, and cooking instructions for a given neuron. Then we use the methodology introduced by Zhou et al. [25] to visualize image regions that contribute the most to the activation of specific units in our learned visual embeddings. We apply the same procedure on the recipe side to also obtain those ingredients and recipe instructions to which certain units react the most. Figure 5 shows the results for the same unit in both the image and recipe embedding. We find that certain units display localized semantic alignment between the embeddings of the two modalities.

Semantic Vector Arithmetic. Different works in the literature [15, 18] have used simple arithmetic operations to demonstrate the capabilities of their learned representations. In the context of food recipes, one would expect that $v(\text{“chicken pizza”}) - v(\text{“pizza”}) + v(\text{“salad”}) = v(\text{“chicken salad”})$, where v represents the map into the embedding space. We investigate whether our learned embeddings have such properties by applying the previous equation template to the averaged vectors of recipes that contain the queried words in their title. We apply this procedure in the image and recipe embedding spaces and show results in Figures 6(a) and 6(b), respectively. Our findings suggest that the learned embeddings have semantic properties that translate to simple geometric transformations in the learned space.

Finally, we apply the same arithmetic operation to embeddings across modalities. In particular, we explore the case of modifying a recipe by linearly combining its image embedding with a variety of text-originated embeddings. For example, given an image of a *chocolate cake*, we try to transform it into a *chocolate cupcake* by removing and adding the mean recipe embeddings of *cake* and *cupcake*, respectively. Figure 6(c) shows the results, which we find to be comparable to those using embeddings within the same modality. This suggests that the recipe and image embeddings learned in our model are semantically aligned, which broaches the possibility of applications in recipe modification (e.g., ingredient replacement, calorie adjustment) or even cross-modal generation.

7. Conclusion

In this paper, we present Recipe1M, the largest structured recipe dataset to date, the im2recipe problem, and neural embedding models with semantic regularization which achieve impressive results for the im2recipe task. More generally, the methods presented here could be gainfully applied to other “recipes” like assembly instructions, tutorials, and industrial processes. Further, we hope that our contributions will support the creation of automated tools for food and recipe understanding and open doors for many less explored aspects of learning such as compositional creativity and predicting visual outcomes of action sequences.

8. Acknowledgements

This work has been supported by CSAIL-QCRI collaboration projects and the framework of projects TEC2013-43935-R and TEC2016-75976-R, financed by the Spanish Ministerio de Economia y Competitividad and the European Regional Development Fund (ERDF).

References

[1] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014. 1, 2

[2] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba. Learning aligned cross-modal representations from weakly aligned data. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016. 5

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 6

[4] V. R. K. Garimella, A. Alfayad, and I. Weber. Social media image analysis for public health. In *CHI*, pages 5543–5547, 2016. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 3

[6] C.-w. N. Jing-jing Chen. Deep-based ingredient recognition for cooking recipe retrival. *ACM Multimedia*, 2016. 2

[7] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 6

[8] Y. Kawano and K. Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, 74(14):5263–5287, 2015. 1, 2

[9] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. In *NIPS*, pages 3294–3302, 2015. 3, 6

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[11] T. Kusmierczyk, C. Trattner, and K. Norvag. Understanding and predicting online food recipe production patterns. In *HyperText*, 2016. 2

[12] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014. 6

[13] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma. Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics*, pages 37–48. Springer, 2016. 1

[14] Y. Mejova, S. Abbar, and H. Haddadi. Fetishizing food in digital age: #foodporn around the world. In *ICWSM*, pages 250–258, 2016. 1

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. 3, 6, 8

[16] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy. Im2calories: Towards an automated mobile vision food diary. In *ICCV*, pages 1233–1241, 2015. 1, 2

[17] F. Ofli, Y. Aytar, I. Weber, R. Hammouri, and A. Torralba. Is saki #delicious? the food perception gap on instagram and its relation to health. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017. 1

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 8

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 3

[21] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014. 3

[22] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 6

[23] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso. Recipe recognition with large multimodal food dataset. In *ICME Workshops*, pages 1–6, 2015. 2

[24] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, and R. Jain. Geolocalized modeling for dish recognition. *IEEE Trans. Multimedia*, 17(8):1187–1199, 2015. 2

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *International Conference on Learning Representations*, 2015. 8

[26] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 1