CookGAN: Meal Image Synthesis from Ingredients

Fangda Han
Rutgers University,
Piscataway, NJ, USA
fh199@cs.rutgers.edu

Ricardo Guerrero
Samsung AI Center, Cambridge, UK
r.guerrero@samsung.com

Vladimir Pavlovic
Rutgers University,
Piscataway, NJ, USA
vladimir@cs.rutgers.edu

Abstract

In this work we propose a new computational framework, based on generative deep models, for synthesis of photo-realistic food meal images from textual list of its ingredients. Previous works on synthesis of images from text typically rely on pre-trained text models to extract text features, followed by generative neural networks (GAN) aimed to generate realistic images conditioned on the text features. These works mainly focus on generating spatially compact and well-defined categories of objects, such as birds or flowers, but meal images are significantly more complex, consisting of multiple ingredients whose appearance and spatial qualities are further modified by cooking methods. To generate real-like meal images from ingredients, we propose Cook Generative Adversarial Networks (CookGAN), CookGAN first builds an attention-based ingredients-image association model, which is then used to condition a generative neural network tasked with synthesizing meal images. Furthermore, a cycle-consistent constraint is added to further improve image quality and control appearance. Experiments show our model is able to generate meal images corresponding to the ingredients.

1. Introduction

Computational food analysis (CFA) has become a pivotal area for the computer vision community due to its real-world implications for nutritional health [3, 22, 11, 4, 2, 23, 13, 1, 15]. For instance, being able to extract food information, including ingredients and calories, from a meal image could help us monitor our daily nutrient intake and manage our diet. In addition to food intake logging, CFA can also be crucial for learning and assessing the functional similarity of ingredients, meal preference forecasting, and computational meal preparation and planning [23, 9]. The advancement of CFA depends on developing better frameworks that aim at extracting food-related information from different domains including text descriptions (*e.g.* recipe title, ingredients, instructions) and meal images, as well as exploring the relationships between different domains in order to better understand food related properties.

This paper concentrates on generating meal images from a set of specific ingredients. Although image generation from text is popular in the computer vision community [20, 28, 27], similar work on generating photo-realistic meal images has so far failed to materialize due to the complex factors associated to meal images, these factors include appearance diversity, dependency on the cooking method, variations in preparation style, visual dissolution, etc. As a consequence, the generative meal model has to infer these key pieces of information implicitly.

In this work, we propose Cook Generative Adversarial Networks (CookGAN), a model to generate a photo-realistic meal image conditioned on a list of ingredients (we will use 'ingredient list' or 'ingredients' interchangeably in this paper). The efficacy of the model is analyzed by modifying the visible ingredients. The main contributions are: 1) Combining attention-based recipe association model [4] and StackGAN [27] to generate meal images from ingredients. 2) Adding a cycle-consistency constraint to further improve image quality and control the appearance of the image by changing ingredients.

2. Related Work

Generative neural networks (GAN) GAN is a popular type of generative model for image synthesis [7]. It learns to model the distribution of real images via a combination of two networks, one that generates images from a random input vector and another that attempts to discriminate between real and generated images.

Conditional GAN Work on generating images conditioned on a deterministic label by directly concatenating the label with the input was proposed by [16] and by adding the label information at a certain layer's output in [18, 17]. Another line of work conditions the generation process with text information. [20] uses a pre-trained model to extract text features and concatenate them with the input random

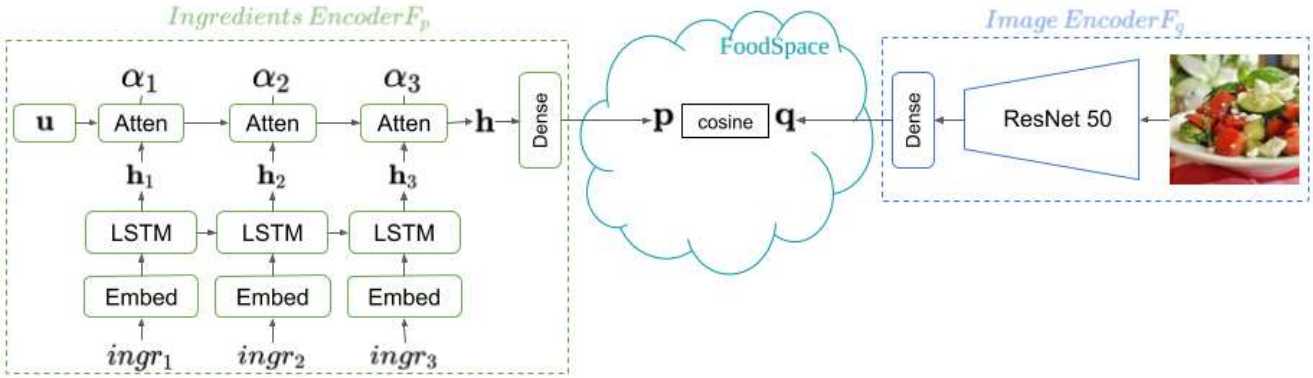

Figure 1: The framework of the attention-based cross-modal association model.

vector, in order to generate text-based images. [27] extends this concept by stacking three GAN to generate images at different resolutions. These works are conditioned on short textual descriptions of the image and rely on recurrent neural networks (RNN) to extract text features. RNNs treat words sequentially and with the same importance. However, in the sparse list of ingredients of a meal, not all ingredients occupy equally important roles in image appearance. Inspired by [4], we combine attention mechanism with bi-directional LSTM (a commonly used RNN model) to learn the importance of each ingredient. The attention mechanism helps locate key ingredients in an unsupervised way.

Meal Image Generation Most prior work for image generation from text implicitly assume the visual categories are well-structured singular objects, consistent in appearance (*e.g*. birds, flowers, or faces). Meal images, on the other hand, have more variable appearance when conditioned on ingredients. [24] and [29] use generative neural networks to generate meal images as a constraint to improve cross-modal recipe retrieval, however, they only generate low-resolution (*e.g*. 128×128) images, and furthermore, because the synthesized image is only used to regularize the retrieval model, image quality is not well evaluated. [19] uses GAN to generate pizza images given step-by-step procedures, however, their model is only tested with pizza with a pre-defined list of procedures and ingredients. Compared with them, we aim at generating meal images with various food types and ingredients.

[25] and [5] are more closely related to our work, however, we include a cycle-consistency regularizer to minimize the semantic discrepancy between fake and real images. The guiding intuition is that if the generated image is of high quality and captures the ingredients correctly, it should extract similar feature as that from the real image. Experiment shows this regularizer improves image quality both qualitatively and quantitatively.

3. Methodology

To generate a meal image from an ingredient list, we first train an attention-based association model to find a shared latent space between ingredient list and image, then use the latent representation of the ingredient list to train a GAN to synthesize the meal image conditioned on the list.

3.1. Attention-based Association Model

In order to extract ingredient features, we train an attention-based cross-modal association model [4] to match an ingredient list and its corresponding image in a joint latent space, denoted the `FoodSpace`. During training, the model takes a triplet as input, which includes the recipe ingredient list, its corresponding image, and an image from another recipe, $(\boldsymbol{r}^+, \boldsymbol{v}^+, \boldsymbol{v}^-)$, respectively. Using two separate neural networks, one for the ingredient list $\mathrm{F_p}$ and another for images $\mathrm{F_q}$, the triplet is embedded in the `FoodSpace` with coordinates $(\boldsymbol{p}^+, \boldsymbol{q}^+, \boldsymbol{q}^-)$. The networks are trained to maximize the association in `FoodSpace` between positive pair $(\boldsymbol{p}^+, \boldsymbol{q}^+)$, and at the same time minimizing the association between negative pair $(\boldsymbol{p}^+, \boldsymbol{q}^-)$.

Formally, with the ingredients encoder $\boldsymbol{p} = \mathrm{F_p}(\boldsymbol{r})$ and image encoder $\boldsymbol{q} = \mathrm{F_q}(\boldsymbol{v})$, the training is a maximization of the following objective function,

$$
\begin{aligned}
&V(\mathrm{F_p}, \mathrm{F_q}) = \\
&\mathbb{E}_{\hat{p}(\boldsymbol{r}^+, \boldsymbol{v}^+), \hat{p}(\boldsymbol{v}^-)} \min\left(\left[d\big[\boldsymbol{p}^+, \boldsymbol{q}^+\big] - d\big[\boldsymbol{p}^+, \boldsymbol{q}^-\big] - \epsilon\right], 0\right) + \\
&\mathbb{E}_{\hat{p}(\boldsymbol{r}^+, \boldsymbol{v}^+), \hat{p}(\boldsymbol{r}^-)} \min\left(\left[d\big[\boldsymbol{p}^+, \boldsymbol{q}^+\big] - d\big[\boldsymbol{p}^-, \boldsymbol{q}^+\big] - \epsilon\right], 0\right),
\end{aligned}
\tag{1}
$$

where $\cos[\boldsymbol{p}, \boldsymbol{q}] = \boldsymbol{p}^\intercal \boldsymbol{q} / \sqrt{(\boldsymbol{p}^\intercal \boldsymbol{p})(\boldsymbol{q}^\intercal \boldsymbol{q})}$ is the cosine similarity in `FoodSpace` and $\hat{p}$ denotes the corresponding empirical densities on the training set. We combine the cosine similarity of the positive pair and that of the negative pair together, and we add a margin ϵ to make the model focus

on those pairs that are not correctly embedded. We empirically set ϵ to 0.3 by cross-validation. Fig. 1 shows a diagram of the attention-based association model. The details of ingredients encoder $\mathrm{F_p}$ and image encoder $\mathrm{F_q}$ are explained below.

Ingredients encoder $\mathrm{F_p}$ takes the recipe's ingredients as input and outputs their feature representation in `FoodSpace`. The goal is to find the embedding that reflect dependencies between ingredients, which could facilitate implicit associations even when some ingredients are not visually observable. For this purpose, the model first embeds the one-hot vector of each ingredient into a low-dimension vector ($ingr_i \in \mathbb{R}^{300}$) using a word2vec model [14], treating the vectors as a sequence input of a bi-directional LSTM[1]. Instead of using the output of the last layer as the output of the LSTM, each hidden state $\boldsymbol{h}_i \in \mathbb{R}^{300}$ is used as the feature of the corresponding ingredient.

As not all ingredients play equally important roles in image appearance, we apply attention mechanism to model the contribution of each ingredient. During training, the model learns a shared contextual vector $\boldsymbol{u} \in \mathbb{R}^{300}$ of the same dimension as the hidden state, and $\boldsymbol{u}$ is then used to assess the attention of each ingredient,

$$\{\alpha_i\} = \mathrm{softmax}\left\{\boldsymbol{u}^T \cdot \boldsymbol{h}_i\right\},\ i \in [1, N], \tag{2}$$

where N is the number of ingredients in the recipe. The attention-based output of LSTM is a weighted summation of all hidden states, $\boldsymbol{h} = \sum_{i=1}^{N} \alpha_i \boldsymbol{h}_i$. The contextual vector $\boldsymbol{u}$ is optimized as a parameter during training and fixed during testing. Our intuition is $\boldsymbol{u}$ can attend on certain ingredients that appear in a specific ingredient list by learning from the training data. Finally, $\boldsymbol{h}$ is projected to `FoodSpace` to yield the ingredients feature $\boldsymbol{p} \in \mathbb{R}^{1024}$.

Image encoder $\mathrm{F_q}$ takes a meal image as input and outputs a feature representing the image in `FoodSpace`. Resnet50 [8] pre-trained on ImageNet [6] is applied as the base model for feature extraction. In order to get a more meaningful feature of the image, we follow [4] and finetune the network on UPMC-Food-101 [26], we use the activation after the average pooling ($\mathbb{R}^{2048}$) and project it to `FoodSpace` to get $\boldsymbol{q} \in \mathbb{R}^{1024}$.

3.2. Generative Meal Image Network

Generative meal image network takes the ingredient list as input and generates the corresponding meal image. The base model StackGAN-v2 [27] contains three branches stacked together. Each branch is responsible for generating image at a specific scale and each branch has its own discriminator which is responsible for distinguish the image at that scale. The framework is shown in Fig. 2.

[1]Hence, we assume a chain graph can approximate arbitrary ingredient dependencies within a recipe.

Generator: The ingredients $\boldsymbol{r}^+$ are first encoded using the pre-trained $\mathrm{F_p}$ (fixed during training StackGAN-v2) to obtain text feature $\boldsymbol{p}^+$. Subsequently, $\boldsymbol{p}^+$ is forwarded through a conditional augmentation network F_{ca} to estimate the distribution $p(\boldsymbol{c}|\boldsymbol{p}^+)$ of the ingredient appearance factor $\boldsymbol{c}$, modeled as the Gaussian distribution

$$\left(\mu_{\boldsymbol{p}^+}, \Sigma_{\boldsymbol{p}^+}\right) = F_{ca}(\boldsymbol{p}^+), \tag{3}$$

$$\boldsymbol{c} \sim p(\boldsymbol{c}|\boldsymbol{p}^+) = \mathcal{N}(\mu_{\boldsymbol{p}^+}, \Sigma_{\boldsymbol{p}^+}), \tag{4}$$

where $\mu_{\boldsymbol{p}^+}$ and $\Sigma_{\boldsymbol{p}^+}$ are the mean and the covariance given the ingredients encoding $\boldsymbol{p}^+$ in `FoodSpace`. This sampling process introduces noise to $\boldsymbol{p}^+$, making the model robust to small perturbations in `FoodSpace`. Variational regularization [12] is applied during training to make $p(c|\boldsymbol{p}^+)$ close to the standard Gaussian distribution,

$$\mathcal{L}_{ca} = D_{KL}\left[\mathcal{N}(\mu_{\boldsymbol{p}^+}, \Sigma_{\boldsymbol{p}^+}) || \mathcal{N}(\mathbf{o}, \boldsymbol{I})\right]. \tag{5}$$

Subsequently, $\boldsymbol{c}$ is augmented with Gaussian noise $\boldsymbol{z} \sim \mathcal{N}(\mathbf{o}, \boldsymbol{I})$ to generate the latent feature $\boldsymbol{h}_0 = F_0(\boldsymbol{z}, \boldsymbol{c})$ for the first branch and the low-resolution image $\tilde{\boldsymbol{v}}_0^+ = T_0(\boldsymbol{h}_0)$, where F_0 and T_0 are modeled by neural networks. Similarly, the medium and high resolution images are generated by utilizing the hidden feature of the previous branches, $\boldsymbol{h}_1 = F_1(\boldsymbol{h}_0, \boldsymbol{c})$, $\tilde{\boldsymbol{v}}_1^+ = T_1(\boldsymbol{h}_1)$ and $\boldsymbol{h}_2 = F_2(\boldsymbol{h}_1, \boldsymbol{c})$, $\tilde{\boldsymbol{v}}_2^+ = T_2(\boldsymbol{h}_2)$. Overall, the generator contains three branches, each responsible for generating the image at a specific scale, $G_0 = \{F_0, T_0\}$, $G_1 = \{F_1, T_1\}$, $G_2 = \{F_2, T_2\}$. Optimization of the generator will be described after introducing the discriminators.

Discriminator: Each discriminator's task is three-fold: (1) Classify real, 'correctly-paired' $\boldsymbol{v}^+$ with ingredient appearance factor $\boldsymbol{c}$ as real; (2) Classify real, 'wrongly-paired' $\boldsymbol{v}^-$ with $\boldsymbol{c}$ as fake; and (3) Classify generated image $\tilde{\boldsymbol{v}}^+$ with $\boldsymbol{c}$ as fake. Formally, we seek to minimize the cross-entropy loss

$$\begin{aligned}\mathcal{L}_i^{cond} = &- \mathbb{E}_{\boldsymbol{v}^+ \sim p_{d_i}}[\log D_i(\boldsymbol{v}^+, \boldsymbol{c})] \\ &+ \mathbb{E}_{\boldsymbol{v}^- \sim p_{d_i}}[\log D_i(\boldsymbol{v}^-, \boldsymbol{c})] \\ &+ \mathbb{E}_{\tilde{\boldsymbol{v}}^+ \sim p_{G_i}}[\log D_i(\tilde{\boldsymbol{v}}^+, \boldsymbol{c})],\end{aligned} \tag{6}$$

where p_{d_i}, p_{G_i}, G_i and D_i correspond to the real image distribution, fake image distribution, generator branch, and the discriminator at the i^{th} scale. To further improve the quality of the generated image, we also minimize the unconditional image distribution as

$$\begin{aligned}\mathcal{L}_i^{uncond} = &- \mathbb{E}_{\boldsymbol{v}^+ \sim p_{d_i}}[\log D_i(\boldsymbol{v}^+)] \\ &- \mathbb{E}_{\boldsymbol{v}^- \sim p_{d_i}}[\log D_i(\boldsymbol{v}^-)] \\ &+ \mathbb{E}_{\tilde{\boldsymbol{v}}^+ \sim p_{G_i}}[\log D_i(\tilde{\boldsymbol{v}}^+)]\end{aligned} \tag{7}$$

Losses: During training, the generator and discriminators are optimized alternatively by maximizing and minimizing (6) and (7) respectively. All generator branches

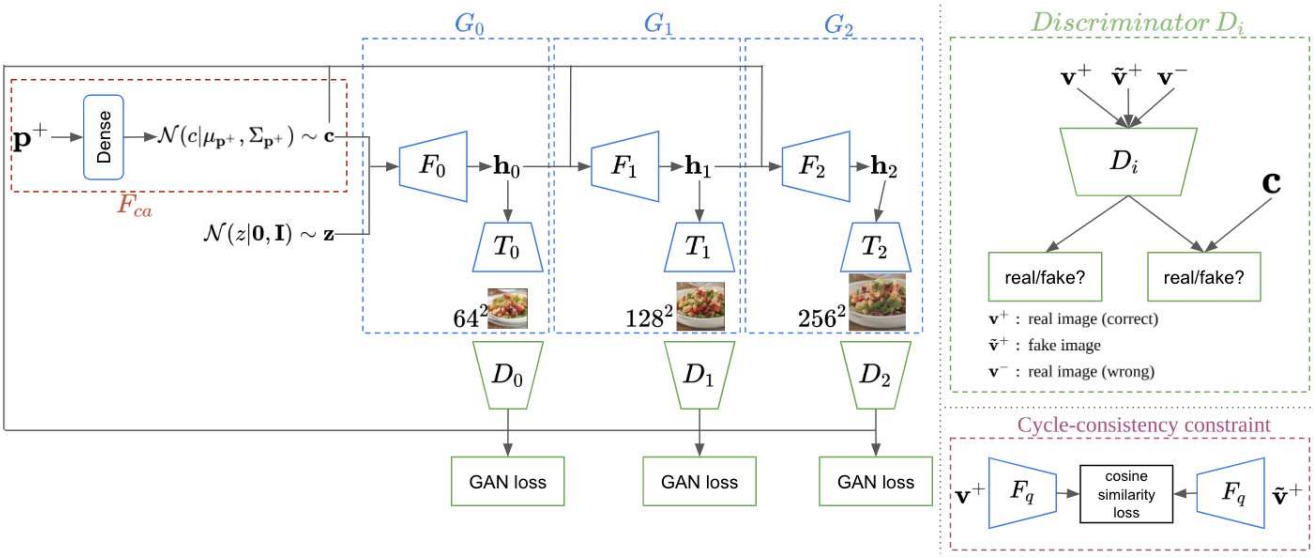

Figure 2: Framework of the generative model. G_0, G_1, G_2 represent the branches in generator. D_0, D_1, D_2 represent the discriminators for images of low, medium and high resolution. F_q is the image encoder trained in the association model.

are trained jointly as are the three discriminators, with final losses

$$\mathcal{L}_G = \sum_{i=0}^{2} \left\{ \mathcal{L}_i^{cond} + \lambda_{uncond} \mathcal{L}_i^{uncond} \right\} + \lambda_{ca} \mathcal{L}_{ca} \quad (8)$$

$$\mathcal{L}_D = \sum_{i=0}^{2} \left\{ \mathcal{L}_i^{cond} + \lambda_{uncond} \mathcal{L}_i^{uncond} \right\}, \quad (9)$$

where λ_{uncond} is the weight of the unconditional loss and λ_{ca} the weight of the conditional augmentation loss. We empirically set $\lambda_{uncond} = 0.5$ and $\lambda_{ca} = 0.02$ by cross-validation.

3.3. Cycle-consistency constraint

A correctly-generated meal image should "contain" the ingredients it is conditioned on. Thus, a cycle-consistency term is introduced to keep the fake image contextually similar, in terms of ingredients, to the corresponding real image in `FoodSpace`.

Specifically, for a real image $\boldsymbol{v}^+$ with `FoodSpace` coordinate $\boldsymbol{q}^+$ and the corresponding generated $\tilde{\boldsymbol{v}}^+$ with $\tilde{\boldsymbol{q}}^+$, the cycle-consistency regularization aims at maximizing the cosine similarity at different scales, $\mathcal{L}_{C_i} = \cos[\boldsymbol{q}^+, \tilde{\boldsymbol{q}}^+]$. Note that the images in different resolutions need to be rescaled for the input of the image encoder. The final generator loss in (8) now becomes

$$\mathcal{L}_G = \sum_{i=0}^{2} \left\{ \mathcal{L}_i^{cond} + \lambda_{uncond} \mathcal{L}_i^{uncond} - \lambda_{cycle} \mathcal{L}_{C_i} \right\} + \lambda_{ca} \mathcal{L}_{ca}, \quad (10)$$

where λ_{cycle} is the weight of the cycle-consistency term, cross-validated to $\lambda_{cycle} = 1.0$.

4. Experiments

Dataset Data used in this work was taken from Recipe1M [22]. This dataset contains ∼1M recipes with titles, instructions, ingredients, and images. We focus on a subset of 402 760 recipes with at least one image, containing no more than 20 ingredients or instructions, and no less than one ingredient and instruction. Data is split into 70% train, 15% validation and 15% test sets, using at most 5 images from each recipe.

Recipe1M contains ∼16k unique ingredients, we reduce this number by focusing on the 4k most frequent ones. This list is further reduced by first merging the ingredients with the same name after a stemming operation and semi-automatically fusing other ingredients. The later is achieved using a word2vec model trained on Recipe1M, where the ingredients are fused if they are close together in their embedding space and a human annotator accepts the proposed merger. Finally, we obtain a list of 1989 canonical ingredients, covering more than 95 % of all recipes in the dataset.

Implementation Details Specific network structures follow those in [4] for the association model[2] and [27] for the generator.

		im2recipe				recipe2im			
		MedR↓	R@1↑	R@5↑	R@10↑	MedR↓	R@1↑	R@5↑	R@10↑
1K	attention [4]	-	-	-	-	-	-	-	-
	ours w/o attn	**5.400**	0.229	**0.510**	**0.621**	**5.736**	0.230	**0.502**	0.610
	ours w/ attn	5.500	**0.234**	0.503	0.618	5.750	**0.230**	0.491	**0.615**
5K	attention [4]	71.000	0.045	0.135	0.202	70.100	0.042	0.133	0.202
	ours w/o attn	**24.000**	**0.105**	0.260	0.360	25.300	0.094	**0.261**	**0.358**
	ours w/ attn	**24.000**	0.099	**0.265**	**0.364**	**25.100**	**0.097**	0.259	0.357
10K	attention [4]	-	-	-	-	-	-	-	-
	ours w/o attn	**47.500**	0.065	0.183	**0.270**	48.500	0.061	**0.189**	**0.272**
	ours w/ attn	47.700	**0.065**	**0.185**	0.267	**48.300**	**0.061**	0.178	0.261

Table 1: Comparison with the baseline [4] for using image as query to retrieve recipe and vice versa. 'w/o attn' means without attention, 'w/ attn' means with attention. '↓' means the lower the better, '↑' means the higher the better, '-' stands for score not reported in [4].

Figure 3: Sample results of using ingredients as query to retrieve images on a 1K dataset. Left: query ingredients. Right: top 5 retrieved images (sorted). Corresponding image is indicated by the green box.

4.1. Effect of Canonical Ingredients

To evaluate the effect of the proposed canonical ingredients, we compare with the attention-based model [4]. Given a query in one modality, the goal is to retrieve the paired point in the other modality by comparing their similarities in `FoodSpace`. The association model is trained on four Tesla K80 for 16 hours (25 epochs) until convergence.

Metrics. We applied the same metrics as [4], including the median retrieval rank (MedR) and the recall at top K (R@K). MedR is computed as the median rank of the true positive over all queries, a lower MedR $\geq$1.00 suggests better performance. R@K computes the fraction of true positives recalled among the top-K retrieved candidates, it is a value between 0 to 100 with the higher score indicating better performance.

Results. In Tab. 1, we report the scores of the baseline model [4] and that of the same model with our canonical ingredients (with and without attention). The performance is greatly improved on 5K samples, which clearly shows the advantage of using our canonical ingredients instead of the raw ingredients data. Fig. 3 illustrates the top 5 retrieved images using the ingredients as the query. Although the correct images do not always appear in the first position, the retrieved images largely belong to the same food type, suggesting commonality in ingredients.

4.2. Effect of Attention

To evaluate the effect of the attention mechanism mentioned in Sec. 3.1, we report the performance of our model for retrieval with or without attention. Interestingly, our model with attention does not achieve better performance. This is somewhat counter-intuitive since it can be seen in Fig. 4 that the model with attention tends to focus on visually important ingredients. For example, in top-left recipe, the model attends on green beans and chicken soup; in top-right recipe, the model attends on mushroom and leeks. It should be noted that the model does not simply attend on ingredients that appears more frequently in the dataset (*e.g.* olive_oil, water, butter) but learns to focus on the ingredients that are more visible for the recipe. We suspect the reason that attention mechanism does not improve to the performance scores is that the RNN model learns the importance of each ingredient implicitly. Nevertheless, the attention mechanism can exist as an unsupervised method to locate important ingredients for a recipe.

[2]Here we use LSTM instead of GRU because they have similar performance as stated in their paper.

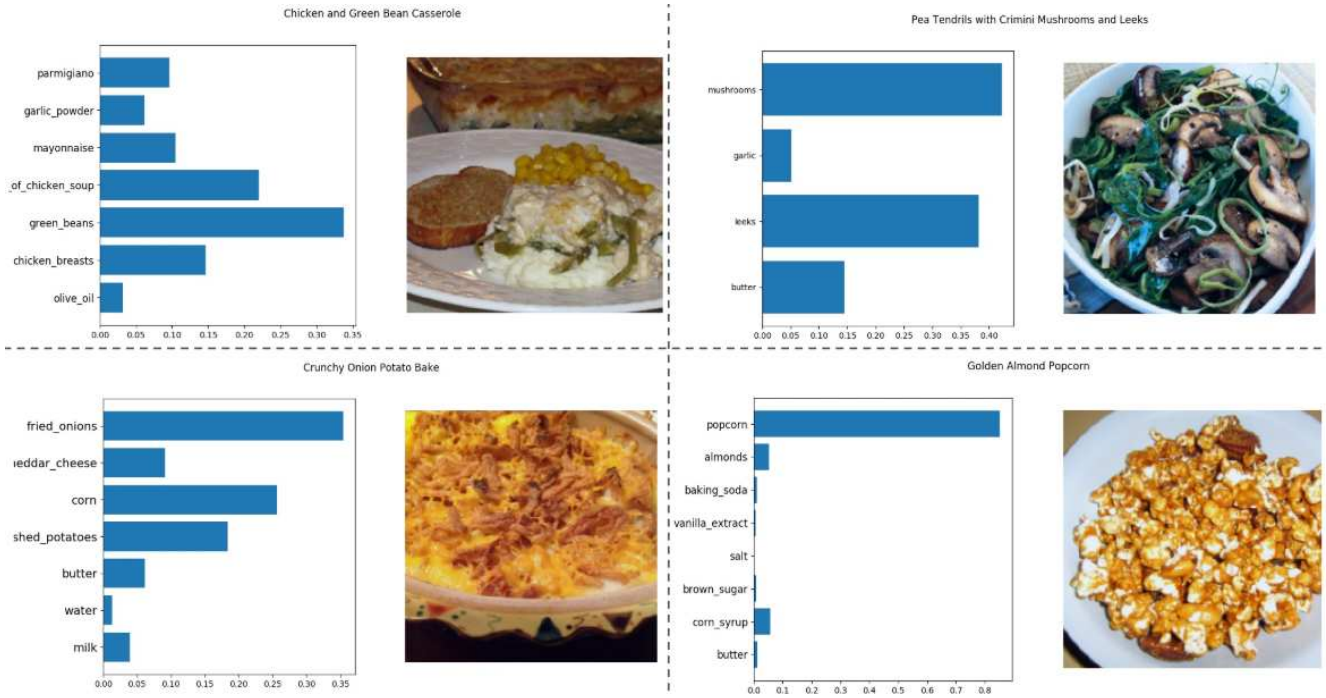

Figure 4: Attention of the ingredients.

		salad	cookie	muffin
IS ↑	StackGAN-v2	3.07	**4.70**	2.60
	ours w/ CI	**3.46**	2.82	**2.94**
	ours w/o CI	3.29	3.53	2.79
	real	5.12	5.70	4.20
FID ↓	StackGAN-v2	**55.43**	106.14	104.73
	ours w/ CI	78.79	87.14	**81.13**
	ours w/o CI	62.63	**89.33**	80.22

(a) Inception score (IS) and Frechet inception distance (FID).

	salad	cookie	muffin
random	450.00	450.00	450.00
StackGAN-v2	**58.40**	194.45	217.50
ours w/ CI	66.15	**103.30**	**211.00**
ours w/o CI	82.42	125.23	232.30
real	12.15	47.35	65.00

(b) Median rank comparison.

Table 2: Performance analysis: (a) Comparison of StackGAN-v2 and our model on different subsets by inception scores (IS) and Frechet inception distance (FID). (b) Comparison of median rank (MedR) by using synthesized images to retrieve recipes in subsets. We choose 900 as the retrieval range to adhere to the maximum number of recipes among test-sets for salad, cookie and muffin. 'w/ CI' means with canonical ingredients, 'w/o CI' means without canonical ingredients.

4.3. Meal Image Generation

We present the results of synthesizing meal image given an ingredient list. To mitigate the diversity caused by different preparation methods, we focus on narrow meal categories where the cutting and cooking methods are largely consistent within each category. In the following experiments, we only train on specific types of food within three commonly-seen categories: salad, cookie, and muffin. Images from these categories usually contain key ingredients that are easily recognized, which can be used to verify the model's ability to manipulate meal image by changing those ingredients. The number of samples in train/test dataset are $17209/3784$ (salad), $9546/2063$ (cookie) and $4312/900$ (muffin).

Metrics. Evaluating the performance of synthesized images is generally a challenging task due to the high complexity of images. We choose Inception Score (IS) [21] and Frechet Inception Distance (FID) [10] as our quantitative evaluation metrics.

Results. We computed IS and FID on 900 samples ran-

Figure 5: Example results by StackGAN-v2 [27] and our model conditioned on target ingredients, the real images are also shown for reference.

Figure 6: Example results from different ingredients **c** with same random vector **z** in the salad subset.

domly generated on the test-set for each category, which is the maximum number of recipes among test-sets for salad, cookie and muffin. The IS of real images are also computed as a baseline. Tab. 2a shows the results obtained on different categories. We compare with StackGAN-v2 [27], one of the state-of-the-art GAN model for text-based image synthesis. **ours w/o CI** uses the original ingredients and the proposed cycle-consistency constraint, while **ours w/ CI** uses the canonical ingredients and the proposed cycle-consistency constraint. We observe the model achieves better IS and FID on most subsets by using cycle-consistency constraint. However, using canonical ingredients does not always lead to better scores for the generative model. We argue that image quality is more related to the design of the generative model while the canonical ingredients help more on the conditioning on the text.

To evaluate the conditioning on the text, we investigate the median rank (MedR) by using synthesized images as the query to retrieve recipes with the association model in Sec. 3.1. Tab. 2b suggests using cycle-consistency constraint outperforms the baseline StackGAN-v2 [27] on most subsets, indicating the utility of the ingredient cycle-consistency. We also observe that applying canonical ingredients always leads to better MedR which demonstrates the effectiveness of our canonical-ingredients-based text embedding model. Still, the generated images remain apart from the real images in their retrieval ability, affirming the extreme difficulty of the photo-realistic meal image synthesis task.

Fig. 5 shows examples generated from different subsets. Within each category, the generated images capture the main ingredients for different recipes. Compared with StackGAN-v2 [27], the images generated using cycle-consistency constraint usually have more clear ingredients appearance and looks more photo-realistic.

Figure 7: Example results from same ingredients with different random vectors. 8 synthesized images are shown for each real image (top-left).

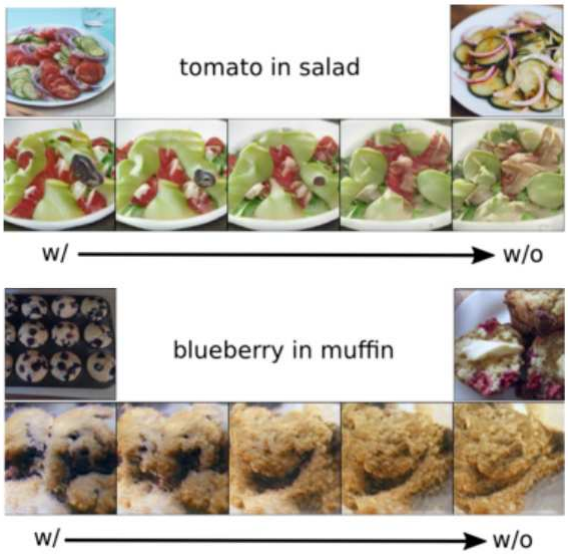

Figure 8: Example results of synthesized images from the linear interpolations in FoodSpace between two recipes (with and without target ingredient). Target ingredient on the left is tomato and the model is trained with salad subset; target ingredient on the right is blueberry and the model is trained with muffin subset. The interpolation points from left to right are $\frac{with}{without} = \{\frac{4}{0}, \frac{3}{1}, \frac{2}{2}, \frac{1}{3}, \frac{0}{4}\}$

4.4. Components Analysis

Our generative model in Fig. 2 has two inputs, an ingredients feature $\mathbf{c}$ and a random vector $\mathbf{z}$. In this section we analyze the different roles played by these two components.

Fig. 6 shows examples generated from different ingredients with the same random vector $\boldsymbol{z}$ in the salad subset. the generated images contains different ingredients for different recipes while sharing a similar view point. This demonstrates the model's ability to synthesize meal images conditioned on ingredient features $\boldsymbol{c}$ while keeping nuisance factors fixed through vector $\boldsymbol{z}$.

Fig. 7 further demonstrates the different roles of ingredients appearance $\boldsymbol{c}$ and random vector $\boldsymbol{z}$ by showing examples generated from same ingredients with different random vectors. The synthesized images have different view points, but still all appear to share the same ingredients.

To demonstrate the ability to synthesize meal images corresponding to specific key ingredient, we choose a target ingredient and show the synthesized images of linear interpolations between a pair of ingredient lists r_i and r_j (in the feature space), in which r_i contains the target ingredient and r_j is without it, but shares at least $70\,\%$ of remaining ingredients in common with r_i[3]. One can observe that the model gradually removes the target ingredient during the interpolation-based removal process, as seen in Fig. 8.

5. Conclusion

In this paper, we develop a model for generating photo-realistic meal images based on sets of ingredients. We integrate the attention-based recipe association model with StackGAN-v2, aiming for the association model to yield the ingredients feature close to the real meal image in FoodSpace, with StackGAN-v2 attempting to reproduce this image class from the FoodSpace encoding. To improve the quality of generated images, we reuse the image encoder in the association model and design an ingredient cycle-consistency regularization term in the shared space. Finally, we demonstrate that processing the ingredients into a canonical vocabulary is a critical key step in the synthesis process. Experimental results demonstrate that our model is able to synthesize natural-looking meal images corresponding to desired ingredients, both visually and quantitatively, through retrieval metrics. In the future, we aim at adding additional information including recipe instructions and titles to further contextualize the factors such as the meal preparation, as well as combining the amount of each ingredient to synthesize images with arbitrary ingredients quantities.

[3]The reason for choosing the partial overlap is because very few recipes differ in exactly one key ingredient.

References

[1] E. Aguilar, B. Remeseiro, M. Bolaños, and P. Radeva. Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia*, 20(12):3266–3275, 2018.

[2] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1:196, 2011.

[3] M. Carvalho, R. Cadène, D. Picard, L. Soulier, N. Thome, and M. Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 35–44. ACM, 2018.

[4] J.-J. Chen, C.-W. Ngo, F.-L. Feng, and T.-S. Chua. Deep understanding of cooking procedure for cross-modal recipe retrieval. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1020–1028. ACM, 2018.

[5] O. B. El, O. Licht, and N. Yosephian. Gilt: Generating images from long text. *arXiv preprint arXiv:1901.02404*, 2019.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] T. Helmy, A. Al-Nazer, S. Al-Bukhitan, and A. Iqbal. Health, food and user's profile ontologies for personalized information retrieval. *Procedia Computer Science*, 52:1071–1076, 2015.

[10] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[11] D. Horita, R. Tanno, W. Shimoda, and K. Yanai. Food category transfer with conditional cyclegan and a large-scale food image dataset. In *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, pages 67–70. ACM, 2018.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[13] N. Martinel, G. L. Foresti, and C. Micheloni. Wide-slice residual networks for food recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 567–576. IEEE, 2018.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[15] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain. A survey on food computing. *CoRR*, abs/1808.07202, 2018.

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[17] T. Miyato and M. Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[18] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[19] D. P. Papadopoulos, Y. Tamaazousti, F. Ofli, I. Weber, and A. Torralba. How to make a pizza: Learning a compositional layer-based gan model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8002–8011, 2019.

[20] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[21] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[22] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3020–3028, 2017.

[23] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 298–307. ACM, 2012.

[24] H. Wang, D. Sahoo, C. Liu, E.-p. Lim, and S. C. Hoi. Learning cross-modal embeddings with adversarial networks for cooking recipes and food images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11572–11581, 2019.

[25] S. Wang, H. Gao, Y. Zhu, W. Zhang, and Y. Chen. A food dish image generation framework based on progressive growing gans. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 323–333. Springer, 2019.

[26] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso. Recipe recognition with large multimodal food dataset. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[27] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017.

[28] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.

[29] B. Zhu, C.-W. Ngo, J. Chen, and Y. Hao. R2gan: Cross-modal recipe retrieval with generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11477–11486, 2019.