Cross-Modal Recipe Embeddings by Disentangling Recipe Contents and Dish Styles

Yu Sugiyama   Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
{sugiya-y,yanai}@mm.inf.uec.ac.jp

ABSTRACT

Nowadays, cooking recipe sharing sites on the Web are widely used, and play a major role in everyday home cooking. Since cooking recipes consist of dish photos and recipe texts, cross-modal recipe search is being actively explored. To enable cross-modal search, both food image features and cooking text recipe features are embedded into the same shared space in general. However, in most of the existing studies, a one-to-one correspondence between a recipe text and a dish image in the embedding space is assumed, although an unlimited number of photos with different serving styles and different plates can be associated with the same recipe.

In this paper, we propose a RDE-GAN (Recipe Disentangled Embedding GAN) which separates food image information into a recipe image feature and a non-recipe shape feature. In addition, we generate a food image by integrating both the recipe embedding and a shape feature. Since the proposed embedding is free from serving and plate styles which are unrelated to cooking recipes, the experimental results showed that it outperformed the existing methods on cross-modal recipe search. We also confirmed that only either shape or recipe elements can be changed at the time of food image generation.

CCS CONCEPTS

• **Information systems** → **Multimedia databases**; **Multimedia content creation**.

KEYWORDS

cross-modal search, recipe embedding, feature disentanglement, GAN

ACM Reference Format:
Yu Sugiyama   Keiji Yanai. 2021. Cross-Modal Recipe Embeddings by Disentangling Recipe Contents and Dish Styles. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3474085.3475422

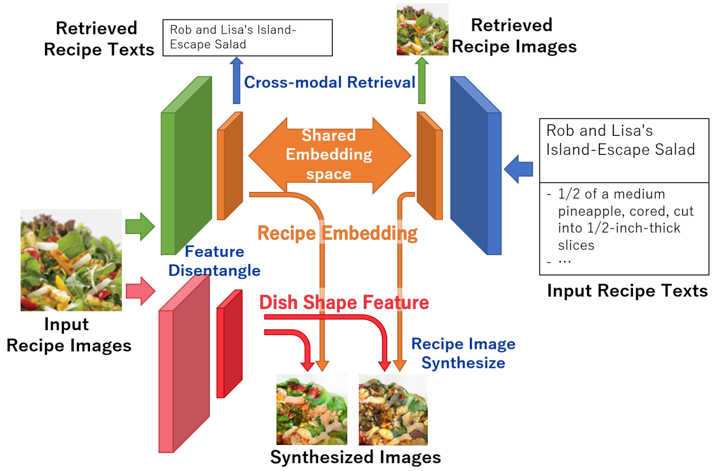

Figure 1: The overview of Recipe Disentangled Embedding GAN (RDE-GAN), where image features are disentangled into a recipe image embedding and a dish shape feature to confirm that the recipe embedding contains only recipe-related information.

1 INTRODUCTION

In these days, cooking recipe sharing sites on the Web, such as Allrecipes and CookPad, are widely used and play a major role in everyday home cooking, since people can search the huge cooking recipe database and get recipe recommendations easily. In addition, any users can post their cooking recipes described in texts and corresponding food photos to the services. So that there are a large number of recipes on the service, and efficient search methods are required such as cross-modal recipe search where text recipes can be searched for from food images and vice versa.

The format of recipe information including texts and images are unified within the same service. Thus, it is easy to collect a large number of pairs of recipe texts and images. The Recipe1M dataset [17, 22] was build by collecting recipe data from 12 cooking recipe sites, containing ingredients and cooking instructions both of which are textual recipe information, and corresponding food photos as visual recipe information. With the advent of this large-scale multi-modal recipe dataset, multi-modal learning of recipe information has become a hot research topic. The Joint Embedding [22] and AdaMine [1] encoded both visual features and textual features of recipe information in the shared space by minimizing the consine distance between them. More recent works, R2GAN [30] and ACME [25], did not only extract embedding vectors but also generated food images from either visual or textual embedding vectors. This helped make the embbedings more informative and improved the accuracy of cross-model recipe search.

However, in all the above-mentioned existing works, they assumed that one dish photo corresponded to one recipe text directly without taking into account the difference of the dish appearances

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475422

which come from the difference of serving styles, plates, viewpoints and lighting conditions. In fact, an unlimited number of food photos corresponding to one certain recipe are possible. Textual recipe information contains the visual information only based on ingredients and cooking styles, and does not contain the information on serving styles, plate shapes and photographing conditions. Therefore, we believe that visual recipe embeddings should not contain such information related to non-recipe factors.

Therefore, in this paper, we propose to disentangle recipe information from non-recipe information, and take account of only recipe information when encoding a dish image into a recipe embedding vector. We expect to improve the performance of cross-modal recipe search by excluding non-recipe information from recipe embeddings.

In this work, as shown in Figure 1, we propose a RDE-GAN (Recipe Disentangled Embedding GAN), which extracts visual and textual cross-modal embeddings for cross-modal search with disentanglement of an image feature into recipe and non-recipe information. RDE-GAN encodes a dish image into two kinds of features, "a visual recipe embedding" and "a dish shape feature." A visual recipe embedding contains recipe information of the dish in the given image, and can be used for cross-modal recipe search, while a dish shape feature contains non-recipe information, which is not used for recipe search. Note that in this paper the visual feature containing only non-recipe information is called "a dish shape feature" for simplicity, although it contains non-recipe information on not only plate shapes but also serving styles, serving amounts, photographing conditions and so on.

As a by-product, we can generate a food image by integrating a recipe embedding and a dish shape feature, which allows us to control the shape of a dish in the generated image. In fact, we can synthesis food images with different recipes on the fixed-shape dish. In addition, since both visual and textual recipe embeddings are embedded in the same share space and interchangeable to each other, we can synthesis food images by combining a textual recipe embedding and a dish shape feature as well. Such food image synthesis can be performed only by the RDE-GAN.

With extensive experiments, we show that our approach is effective and outperforms the existing methods for most of the tasks on cross-modal recipe retrieval. In addition, we show that we achieve new style recipe image generation by integrating recipe embeddings and dish shape features. To summarize it, the contributions of this paper are as follows:

1. We propose to disentangle a recipe image into recipe information and non-recipe information, and use only recipe information to generate visual recipe embeddings. This is the first work of cross-modal recipe embedding which introduces feature disentanglement as long as we know.
2. We propose to generate food images by integrating either a visual or textual recipe embedding which contains only recipe information and a dish shape feature which contains only non-recipe information.
3. We have confirmed the effectiveness of our method by outperforming the existing methods regarding cross-modal recipe search with the Recipe1M dataset [17, 22].
4. We also have confirmed that we can generate novel dish images by integrating a recipe embedding and a dish shape feature extracted from arbitrary food photos and/or recipe texts.

2 RELATED WORK

2.1 Cross-Modal Recipe Retrieval

Recently, cross-modal recipe search has been paid much attention, since cooking recipe Web sites have become very common, and it can be regarded as one of practical applications of multimodal retrieval. The large-scale open recipe multi-modal dataset, Recipe1M [17, 22], has facilitated works on cross-modal recipe search. This dataset contains one million of English cooking recipes each of which consists of both textual information (a title, an ingredients list, and a cooking procedure) and visual information (photos of the cooked foods). In addition, all the recipes have one of 1,047 food categories assigned by the authors, and 4,102 kinds of highly-frequent ingredients are identified as well.

The recipe1M dataset was originally proposed for the first work on cross-model recipe retrieval by the same authors [22]. The authors proposed an image-text cross-modal recipe retrieval method in which both visual features extracted from food photos and textual features extracted from recipe texts are embedded to the shared space. This method is called as Joint Embedding (JE), which showed that it was possible to perform a search beyond the modalities by optimizing the cosine similarity in the shared space.

AdaMine [1] improved retrieval accuracy by learning the shared space using a triplet loss in which a positive pair and a negative pair were taken into account at the same time, while JE adopted a pair-wise loss. Moreover, Adamine introduced selection of hard samples [8] during creation of triplets.

R2GAN [30] and ACME [25] introduced GAN-based image generation in addition to triplet-based joint embedding. R2GAN proposed two-level triplet ranking loss in which triplet was taken into account not only in the shared embedding space but also in the generated image space. In the generated image space, a reconstruct loss was also introduced so that generated images and input dish images become identical. As a result, cross-modal search performance was improved. However, R2GAN intended to improve search performance by introducing image generation rather than to generate high-quality dish images. In fact, they generated 64×64 images, and did not evaluate generated images with quantitative scores such as Fréchet Inception Distance (FID), while ACME [25] and our model generates 128×128 images with the evaluation by FID scores.

ACME [25] performed a reconstruction of ingredients and title category from a visual embedding as well as a recipe image from a textual embedding. ACME adopted adversarial cross-modal training [24] so that a visual embedding made from a recipe image and a textual embedding made from a recipe text cannot be distinguished to each other. The other basic parts are the same as AdaMine [1] and R2GAN [30].

R2GAN [30] and ACME [25] generated food images from only textual recipe embeddings, although recipe texts do not include non-recipe information such as dish shapes and serving styles in general. We expect that this is harmful on the performance of both cross-modal recipe search and food image synthesis. To resolve this problem, we newly introduce feature disentangle of recipe information and non-recipe information in a visual encoder part, and generate food images by integrating both a recipe embedding and a dish shape feature. We prove that this new idea improves the performance on both cross-modal recipe retrieval and food image synthesis by the experiments.

The two recent works on cross-modal recipe embeddings, MCEN [3] and Zan *et al.* [27], focused on improving embeddings without using image generation. MCEN [3] introduced cross-modal attention and consistency to improve recipe retrieval performance. Zan *et al.* [27] introduced attention mechanishm to exclude noise information in food images for reducing the negative effects. In addition, they used BERT [2] as a text encoder. The directions of their approaches on improvements are different from ours. We will integrate these ideas with our framework for future work.

2.2 Feature Disentanglement

Feature disentanglement has been studied for the problem of uninterpretability of latent space in image generation by separating semantic and shape features of an image.

In DRIT [15], a content encoder and a shape encoder were prepared for separation of contents and shapes, and they extracted content features and shape features from an image. By replacing one of the features with one extracted from another input image, they generated an image and encoded it again with the two encoders. Then, the encoders and the generator were trained so that the extracted features become identical to the features provided to the generator.

In MUNIT [12], the network architecture which is based on the arbitrary style transfer network [11] is very unique, although the way of disentanglement training is similar to DRIT. In MUNIT, a global average pooling layer was used to squash spatial features for extracting style features, and style features are injected into a decoder network using Adaptive Instance Normalization Layer (AdaIN) [11]. Image reconstruction was performed based on both content features and style features.

In both the methods, independent encoders were used for encoding of contents and shapes, and they were trained so that the features obtained by re-encoding of a generated image become close to the original encoded features. Regarding feature disentanglement, our work was inspired by DRIT [15] and MUNIT [12]. Our work follows their basic idea of feature disentanglement on contents and shapes.

2.3 Food Image Synthesis

Recently, GAN-based food image generation has been explored. In some works [10, 13], food images were generated by conditional GANs [18] being controlled with food category conditions. The GANs were also used in the two existing cross-modal embedding methods, R2GAN [30] and ACME [25], in which they used cGAN [18] to generate corresponding dish images from either the visual recipe embedding or the textual recipe embedding. In addition, ACME used WGAN-GP [5] to make it indistinguishable if the embedding derived from a recipe image or a recipe text. For image generation in our work we use cGAN and LSGAN, and we also use WGAN-GP for training embedding by following [24].

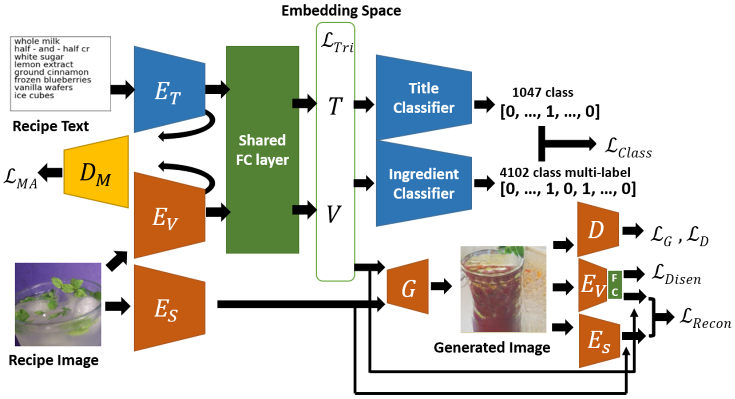

Figure 2: The architecture of RDE-GAN (Recipe Disentangled Embedding GAN), which consists of a recipe text encoder, E_T, a recipe image encoder, E_V, the shared fully-connected (FC) layer, FC, a dish shape encoder, E_S, and an image generator, G.

Recently, food image synthesis from recipe texts has been proposed. CookGAN [29] proposed a cooking simulator sub-network to incrementally change food images based on the interaction between ingredients and cooking methods. The other CookGAN [6] and ChefGAN [20] are similar to each other, which used joint image-recipe embeddings to control the cascaded image generation module based on StackGAN [28]. These works focused on high quality food image synthesis, while the main objective of our work is obtaining cross-modal recipe embeddings for recipe search, and food image synthesis is a by-product. In addition, they cannot control the shape of generated food images unlike our method, since they did not adopt feature disentanglement of recipe features and dish shape features.

3 PROPOSED METHOD

3.1 Overview

Figure 2 shows the basic architecture of the proposed model, RDE-GAN (Recipe Disentangled Embedding GAN). Although we follow ACME [25] regarding the basic structure, the biggest difference is that we disentangle recipe image features and dish shape features to exclude non-recipe information such as serving and plate styles from recipe embeddings. Therefore, our architecture has two kinds of image encoders, a recipe image encoder for extracting recipe factors from a given dish image and a dish shape encoder for extracting non-recipe factors.

As shown in Figure 2, the main elements of the network are a recipe text encoder, E_T, a recipe image encoder, E_V, the shared fully-connected (FC) layer, **FC**, a dish shape encoder, E_S, and an image generator, G. Textual embedding, $T = \mathbf{FC}(E_T(\mathbf{t}))$, and visual embedding, $V = \mathbf{FC}(E_V(\mathbf{i}))$, are obtained in the same shared embedding space, which enables cross-modal recipe search. Note that $\mathbf{t}$ and $\mathbf{i}$ represents a recipe text and a recipe image, respectively. T and V are trained with triplet ranking loss in the same way as AdaMine [1] and ACME [25]. In addition, a recipe category classifier and a multi-label ingredient classifier are used for training. Following [24] and [25], we adopt adversarial cross-modal training with a

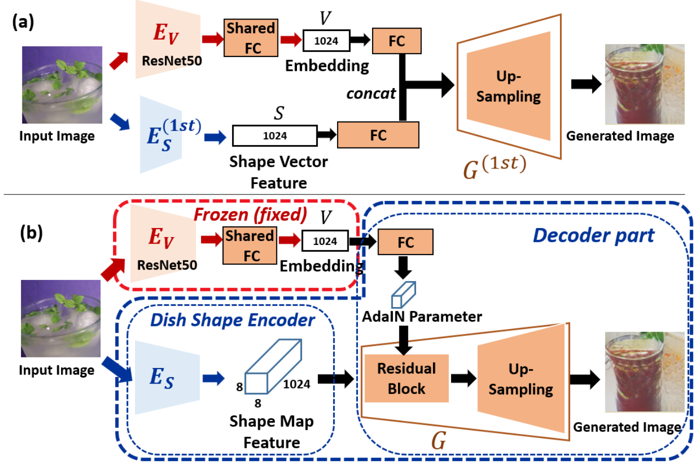

Figure 3: Two-stage training of the image encoders. (a) The first stage: $E_S^{(1st)}$ is a simple down-sampling network and a dish shape feature is a vector. (b) The second stage: The part of E_S and G (marked with blue dotted boxes) are replaced with an encoder which outputs a shape feature map and an AdaIN-style image generator, respectively, while E_V (marked with a red dotted box) is frozen with the parameters trained in the first stage.

modal discriminator, D_M, to make the distributions of outputs of E_T and E_V closer. The dish shape encoder, E_S, encodes non-recipe information as a dish shape feature, S, from a recipe image. For generating recipe images, a generator, G, takes either a text-based embedding, T, or an image-based embedding, V, with a dish shape feature, S, as inputs, and outputs a synthesized image taking account of both recipe information and non-recipe information by adversarial training with an image discriminator, D.

In our method, we train image encoders in two stages to make training process more stable. Figure 3 shows two training stages, (a) the first stage and (b) the second stage, on the part of a recipe image encoder, E_V, a dish shape encoder, E_S, and a generator, G. Note that a recipe text encoder, E_T, is trained in the first stage although it is not shown in this figure.

In the first stage, we mainly focus on training the embedding part of the whole network. To do that, we adopt a simple network for E_S and G. In the second stage, we focus on training the image synthesis part. We replace E_S and G with richer networks and re-train them with the trained embedding part frozen. Since the networks of E_S and G are changed from the first stage to the second stage, we represent E_S and G trained in the first stage as $E_S^{(1st)}$ and $G^{(1st)}$, respectively.

3.2 Cross-Modal Embeddings

A textual embedding, T, and a visual embedding, V, in the shared space are estimated in the same way as ACME [25]. First, we encode both a recipe image and a recipe text to 1024-dimensional semantic vectors with a recipe text encoder, E_T, and a recipe image encoder, E_V. The text encoder, E_T, consists of a bi-directional LSTM encoding ingredients lists, hierarchical LSTMs encoding cooking instructions and a FC layer which integrates the outputs of two kinds of LSTM-based encoders, which are basically the same as that of Joint Embedding (JE) [22]. For image encoding, we use a ResNet50 [7] pre-trained with ImageNet after replacing the final FC layer with a new FC layer as a recipe image encoder, E_V. We embed the two encoded semantic vectors into the shared space by passing through the same fully-connected layer, **FC**, by following ACME and CM-GANs [21].

To train the embedding part, we use three kinds of the loss functions: (1) distance learning loss, (2) modality adversarial loss, and (3) classification loss.

3.2.1 *Triplet-based Distance Learning Loss.*

Distance learning on embeddings is performed by triplet learning [8] with hard samples as in ACME [25]. For a certain pair of textual recipe embeddings and visual embeddings, the corresponding ones are assumed to be positive samples and the ones that do not correspond to each other are assumed to be negative. We select hard ones to separate from many negative samples to perform effective distance learning.

In the triplet learning [23, 26], we take one anchor sample from a training set, select positive one and negative one to the anchor sample, and train the embedding function so that the difference between the distance from the anchor to the negative one and the distance from the anchor to the positive one becomes more than the pre-defined margin.

The triplet learning in the cross-modal embedding takes an anchor and its positive/negative samples from the different modality. We take one anchor from visual embeddings, V_a, and positive/negative samples from textual embeddings, T_p, T_n. At the same time, we take a textual anchor, T_a, and positive/negative visual samples, V_p, V_n, as well. The triplet loss function with the cosine distance d is as follows:

$$\begin{aligned}\mathcal{L}_{Tri} = &\sum_{V}[d(V_a, T_p) - d(V_a, T_n) + \alpha]_+ \\ &+ \sum_{T}[d(T_a, V_p) - d(T_a, V_n) + \alpha]_+ \\ &\text{where } [z]_+ = \max(z, 0).\end{aligned} \tag{1}$$

3.2.2 *Modality Adversarial Loss.*

We prepare a discriminator D_M that discriminates whether embeddings in the shared space come from textual or visual modality, and then we perform adversarial training so that the discriminator cannot discriminate. As a result, the distributions of visual embeddings and textual embeddings are expected to become close to each other. This idea was originally proposed in [24], and ACME adopted it. We use WGAN-GP [5] to perform this adversarial training to prevent mode collapse. The modality adversarial loss can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{MA} =&\mathbb{E}_{\mathbf{i}\sim p(\mathbf{i})}[\log D_M(E_V(\mathbf{i}))] + \\ &\mathbb{E}_{\mathbf{t}\sim p(\mathbf{t})}[\log(1 - D_M(E_T(\mathbf{t})))],\end{aligned} \tag{2}$$

where $\mathbf{i}$, $\mathbf{t}$, $p(\mathbf{i})$ and $p(\mathbf{t})$ represent a recipe image, the corresponding recipe text, the distribution of image samples, and the distribution of recipe text samples, respectively.

3.2.3 *Classification Loss on Recipe Labels and Ingredients.*

Both textual and visual embedding vectors should have recipe information. To confirm it, classification loss on recipe title category and ingredients is introduced. Since each recipe in the Recipe1M dateset has one of 1,047 class labels assigned by hand based on recipe titles, we classify both embeddings into one of 1,047 recipe categories. In addition, we estimate 4,102-d ingredient vectors from them as multi-label classification. We minimize the following loss function with cross-entropy losses by a title classifier and an ingredient classifier, which are represented as $\mathcal{L}_{Title}$ and $\mathcal{L}_{Ingr}$:

$$\begin{aligned}\mathcal{L}_{Class} = \mathcal{L}_{Title}(V, l_T) + \mathcal{L}_{Title}(T, l_T) \\ + \mathcal{L}_{Ingr}(V, l_I) + \mathcal{L}_{Ingr}(T, l_I),\end{aligned} \quad (3)$$

where l_T and l_I represent a groundtruth food category label and a groundtruth ingredient vector, respectively.

3.3 Food Image Synthesis

In the existing works such as ACME and R2GAN, food images are generated from only recipe embeddings, while in our work we disentangle recipe information from non-recipe information, embed only recipe information into recipe embeddings and encode non-recipe information into a dish shape feature. To generate food images, we use both recipe embeddings and dish shape features. This is the biggest difference to the existing methods.

As explained in Section 3.1, the image generator part is trained in two stages. As shown in Figure 3, in (a) the first stage, we use a vector representation for a shape feature, while in (b) the second stage, we use a feature map representation for a shape feature. The reason why we adopted two-stage training is that a shape map feature has too strong representation power which makes it difficult to separate recipe information and non-recipe information using an image recipe encoder, E_V, and a dish shape encoder, E_S, appropriately. In fact, when using a shape map feature from the beginning of the training process, the trained image generator generated dish images only from shape map features ignoring recipe embeddings. We encountered this problem in the early stage of this work. So we intend to focus on training recipe embeddings in the first stage, and in the second stage we focus on training an image generator and a dish shape encoder. The detail will be explained later in this subsection, and the requirement of two-step training will be validated in the section on ablation studies.

In this subsection, we describe the network of each of the two stages, and two kinds of losses, (1) adversarial loss and (2) disentanglement loss, to train the image synthesis part.

3.3.1 Image Synthesis in First Stage. In the first stage, as shown in Figure 3(a), we adopt a simple network for an initial dish shape encoder, $E_S^{(1st)}$, and a dish image generator, $G^{(1st)}$ since we focus on training of the embedding part. We use a 1024-d dish shape vector in the first stage, which is the same dimension as recipe embeddings.

$E_S^{(1st)}$ consists of one 7x7 convolutional layer (conv), four 3x3 convs with stride 2, a global average pooling layer (GAP) and one FC, while $G^{(1st)}$ consists of one FC, five pairs of a nearest-neighbor up-sampling layer and a 3x3 conv, and a hyperbolic tangent function (tanh). Since the first-stage shape encoder has a GAP layer which squashes spatial information, this encoder is not enough to reconstruct a recipe image accurately. So that in the second stage we replace it with the final shape encoder which outputs an 8x8x1024 feature-map-style feature.

3.3.2 Image Synthesis in Second Stage. In the second stage, we focus on training of the image synthesis part, assuming that both the visual and textual embedding encoders are trained successfully in the first stage. Therefore, all the networks except for the image synthesis part (E_S and G) are frozen with the parameters trained in the first stage.

The purpose of this stage is to improve the quality of image generation and the ability of image manipulation by controlling recipe embeddings and shape features. To do that, as shown in Figure 3(b), we discard the initial shape encoder, $E_S^{(1st)}$, and the initial image generator, $G^{(1st)}$, and import the final dish shape encoder, E_S and the final image generator, G. As model architectures of E_S and G in the second stage, we use the Content Encoder in MUNIT [12] as a dish shape encoder, and the Decoder in MUNIT as a recipe image generator. The dish shape encoder outputs a 8x8x1024 feature-map-style feature as a dish shape feature. The recipe image generator uses an Adaptive Instance Normalization layer (AdaIN) [11] to control the style of generated images with a style code. Since AdaIN is used in the state-of-the-art GAN, StyleGAN [14], as well, high-quality images are expected to be generated. In MUNIT, a style code extracted by the Style Encoder was provided directly to the Decoder. Instead in our model we regard a recipe embedding in the shared embedding space as a style code. Since our model is a cross-model embedding model, both visual and textual recipe embeddings can be used as a style code which controls recipe information of the food images generated by the image generator. Note that the size of synthesized images is 128x128, which is the same size as ACME [25].

3.3.3 Adversarial Loss. For training of a generator, we adopt adversarial training using a discriminator as well as a generator. In the first stage for an initial generator, $G^{(1st)}$, we use a vanilla GAN [4] following ACME [25] and R2GAN [30]. In the second stage, we focus on training a generator with the networks of embedding parts frozen. We use LSGAN [16] because we experimentally found that it generated higher-quality images than SNGAN [19] and WGAN-GP [5]. To improve the quality of image generation, we applied the discriminator to the images generated from textual embeddings, T, (txt2img), as well as images generated from visual embeddings, V, (img2img). The loss functions of a discriminator, D, and a generator, G, are as follows:

$$\begin{aligned}
\mathcal{L}_D &= \mathcal{L}_{D_{real}} + \mathcal{L}_{D_{img2img}} + \mathcal{L}_{D_{txt2img}} && (4)\\
&= \mathbb{E}_{\mathbf{i}\sim p(\mathbf{i})}\left[(D(\mathbf{i}) - b)^2\right] \\
&+ \mathbb{E}_{\mathbf{i}\sim p(\mathbf{i})}\left[(D(G(V(\mathbf{i}), E_S(\mathbf{i}))) - a)^2\right], \\
&+ \mathbb{E}_{\mathbf{i}\sim p(\mathbf{i}),\mathbf{t}\sim p(\mathbf{t})}\left[(D(G(T(\mathbf{t}), E_S(\mathbf{i}))) - a)^2\right] \\
\mathcal{L}_G &= \mathcal{L}_{G_{img2img}} + \mathcal{L}_{G_{txt2img}} && (5)\\
&= \mathbb{E}_{\mathbf{i}\sim p(\mathbf{i})}\left[(D(G(V(\mathbf{i}), E_S(\mathbf{i}))) - c)^2\right] && (6)\\
&+ \mathbb{E}_{\mathbf{i}\sim p(\mathbf{i}),\mathbf{t}\sim p(\mathbf{t})}\left[(D(G(T(\mathbf{t}), E_S(\mathbf{i}))) - c)^2\right] && (7)\\
&\quad where\ V(\mathbf{i}) = \mathbf{FC}(E_V(\mathbf{i})), T(\mathbf{t}) = \mathbf{FC}(E_T(\mathbf{t})),
\end{aligned}$$

i and **t** represent an recipe image and the corresponding recipe text in the training set, respectively, and a, b, and c, are the constants in LSGAN [16]. In the experiments, we used $(a, b, c) = (-1, 1, 0)$ following [16].

3.3.4 Disentanglement Loss. We prepared two losses for feature disentanglement of recipe image embeddings and dish shape features, which are (1) reconstruction loss, and (2) disentanglement loss. The basic idea is that we encode two images associated with different recipes, $\mathbf{i_1}$ and $\mathbf{i_2}$, to visual embedding, V_1 and V_2, and dish shape features, $E_S(\mathbf{i_1})$ and $E_S(\mathbf{i_2})$, generate food images by combining them in several ways, and examine (1) if the same embedding or the same shape feature can be extracted from the generated image or (2) if the generated image is classified into the same title category as the recipe title category of the input image.

With the reconstruction loss, we confirm if the original feature can be encoded from the generated images. We generate the image, $\mathbf{i_{fake}}$, with V_1 and $E_S(\mathbf{i_2})$, and examine if V_1 and $E_S(\mathbf{i_2})$ can be reconstructed from $\mathbf{i_{fake}}$. The difference between the original feature and the re-encoded features is evaluated by the L1 distance. This idea is imported from the latent reconstruction loss proposed in MUNIT [12]. The loss function of the reconstruction loss is as follows:

$$\begin{aligned}\mathcal{L}_{Recon} &= \mathbb{E}_{\mathbf{i_1},\mathbf{i_2}\sim p_{data}(\mathbf{i})}\left[|V(\mathbf{i_1}) - \mathbf{FC}(E_V(G(V(\mathbf{i_1}), E_S(\mathbf{i_2}))))|_1\right] \\ &+ \mathbb{E}_{\mathbf{i_1},\mathbf{i_2}\sim p_{data}(\mathbf{i})}\left[|E_S(\mathbf{i_2}) - E_S(G(V(\mathbf{i_1}), E_S(\mathbf{i_2})))|_1\right] \\ &\quad where\ V(\mathbf{i}) = \mathbf{FC}(E_V(\mathbf{i})). \end{aligned} \tag{8}$$

The disentanglement loss is to confirm if a generated image reflects recipe semantic information correctly. We confirm if the generated image is always classified into the same title category and reconstructs the same ingredient list with the fixed visual embedding even when the shape features are changed. To do that, we apply the classification loss (Eq.3) to generated images as a disentanglement loss. The loss function of the disentanglement loss is as follows:

$$\begin{aligned}\mathcal{L}_{Disen} &= \mathcal{L}_{Title}(V'(\mathbf{i_1},\mathbf{i_1}), l_T) + \mathcal{L}_{Title}(V'(\mathbf{i_1},\mathbf{i_2}), l_T) \\ &+ \mathcal{L}_{Ingr}(V'(\mathbf{i_1},\mathbf{i_1}), l_I) + \mathcal{L}_{Ingr}(V'(\mathbf{i_1},\mathbf{i_2}), l_I) \\ &\quad where\ V'(\mathbf{i_1},\mathbf{i_2}) = \mathbf{FC}(E_V(G(V(\mathbf{i_1}), E_S(\mathbf{i_2})))), \\ &\quad V(\mathbf{i}) = \mathbf{FC}(E_V(\mathbf{i})), \end{aligned} \tag{9}$$

and l_T and l_I represent a groundtruth food category label and a groundtruth ingredient vector, respectively.

3.4 Training

By summing all the loss functions explained so far, the total loss function can be written as follows:

$$\begin{aligned}\mathcal{L}_{Total} &= \lambda_1\mathcal{L}_{Class} + \lambda_2\mathcal{L}_{MA} + \lambda_3\mathcal{L}_{Tri} \\ &+ \lambda_4\mathcal{L}_G + \lambda_5\mathcal{L}_{Disen} + \lambda_6\mathcal{L}_{Recon}, \end{aligned} \tag{10}$$

where $\lambda_1, ..., \lambda_6$ are constants weighting losses.

The total loss function is minimized by adversarial training using the discriminators, D_M and D. Training is performed in two stages as explained above. In the first stage, we train all the network including the recipe image encoder, E_V, the recipe text encoder, E_T, a temporary dish shape encoder and generator, $E_S^{(1st)}$ and $G^{(1st)}$, while in the second stage we train a final dish shape encoder and generator, E_S and G, with the recipe image encoder and the recipe text encoder, E_V and E_T, fixed. In both the stages, we train the networks in the end-to-end manner.

4 EXPERIMENTS

We evaluated our proposed model using Recipe1M [17, 22]. Following [22], we used 238,999 image-recipe pairs for training, 51,119 pairs for validation, and 51,303 pairs for testing. We used Adam with initial learning rate at 0.0001 for training. We trained the first-stage model for the first 50 epochs. After that we changed the stage and trained the second-stage model for the next 50 epochs.

We empirically decided the value of each of the loss weights. Basically we adjusted the weighting constants so that each of the losses affected to the total loss equally. The loss weights for Eq.10 used in the experiments are as follows: $\lambda_1 = 0.02$, $\lambda_2 = 0.01$, $\lambda_3 = 1.0$, $\lambda_4 = 0.02$, $\lambda_5 = 0.005$, and $\lambda_6 = 0.5$.

4.1 Evaluation on Cross-modal Recipe Search

We evaluated cross-modal search performance quantitatively regarding both recipe image search from texts (text-to-image search) and recipe text search from images (image-to-text search). The accuracy of the recipe retrieval of the existing methods and the proposed method were evaluated by the MedR and the recall rate. MedR is an index representing the median ordered good search ranking of all data. Recall rate is an index showing that the proportion of correct data presents within 1, 5, or 10-th ranks in the search of all the data. 1,000 or 10,000 samples were randomly sampled from the test set of Recipe1M. We have evaluated a search accuracy on the average of 10 experiments. In the experiments, we used JE [22], AdaMine [1], R2GAN [30], ACME [25], MCEN [3] and Zan *et al.* [27] as baselines. The details of them were explained in Section 2.1.

The results are shown in Table 1. For the larger test set, 10k, on both image-to-text and text-to-image cross-modal search, the proposed method clearly outperformed all the state-of-the-arts regarding all of the statistics with large margin, and for the smaller test set, 1k, it outperformed all the baselines regarding R@1 as well. On the other hand, for R@5 with 1k, it could not outperform Zan *et al.* [27], and for R@10 with 1k it could not outperform Zan *et al.* [27] and ACME [25]. For R@5 and R@10 with the smaller test set, 1k, the performance of the baselines is more than 80%, which means that the room to be improved are relatively smaller than 10k and R@1 of 1k. In fact, the recipe1M dataset contains some noisy texts and noisy images such as photos of human faces and recipe books, since it was created by crawling commercial recipe sites without detailed screening by human. We expect this is one of the possible reasons why it was difficult to improve R@5 and R@10 with 1k, although our method was able to outperform all the baselines on all the scores in case of 10k with large margin.

From these observations, we conclude that disentanglement of recipe information and non-recipe information for encoding visual recipe embeddings helped improve cross-modal search performance effectively.

4.2 Ablation Study

To examine if feature disentanglement helped performance improvement more, we made ablation studies with 10k samples. The functions related to image generator, G, in the total loss function (Eq.10) are an image generator adversarial loss, $\mathcal{L}_G$ (Eq.5), and two disentanglement-related losses, $\mathcal{L}_{Disen}$ (Eq.9) and $\mathcal{L}_{Recon}$ (Eq.8). Note that $\mathcal{L}_G$ consists of $\mathcal{L}_{G_{txt2img}}$ (Eq.7) and $\mathcal{L}_{G_{img2img}}$ (Eq.6),

Table 1: Comparison of the retrieval performance to the baselines.

Test set size		Image-to-Text retrieval				Text-to-Image retrieval			
	Methods	MedR↓	R@1↑	R@5↑	R@10↑	MedR↓	R@1↑	R@5↑	R@10↑
1k	JE [22]	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0
	AdaMine [1]	**1.0**	39.8	69.0	77.4	**1.0**	40.2	68.1	78.7
	R2GAN [30]	2.0	39.1	71.0	81.7	2.0	40.6	72.6	83.3
	MCEN [3]	2.0	48.2	75.8	83.6	1.9	48.4	76.1	83.7
	ACME [25]	**1.0**	51.8	80.2	87.5	**1.0**	52.8	80.2	87.6
	Zan *et al.* [27]	**1.0**	52.7	**81.7**	**88.9**	**1.0**	54.1	**81.8**	**88.9**
	RDE-GAN (ours)	**1.0**	**59.4**	81.0	87.4	**1.0**	**61.2**	81.0	87.2
10k	JE [22]	41.9	-	-	-	39.2	-	-	-
	AdaMine [1]	13.2	14.9	35.3	45.2	12.2	14.8	34.6	46.1
	R2GAN [30]	13.9	13.5	33.5	44.9	12.6	14.2	35.0	46.8
	MCEN [3]	7.2	20.3	43.3	54.4	6.6	21.4	44.3	55.2
	Zan *et al.* [27]	7.0	22.1	45.9	56.9	7.0	23.4	47.3	57.9
	ACME [25]	6.7	22.9	46.8	57.9	6.0	24.4	47.9	59.0
	RDE-GAN (ours)	**3.5**	**36.0**	**56.1**	**64.4**	**3.0**	**38.2**	**57.7**	**65.8**

and $\mathcal{L}_{G_{txt2img}}$ are already used in ACME. So we remove $\mathcal{L}_{G_{img2img}}$, $\mathcal{L}_{Disen}$ and $\mathcal{L}_{Recon}$ to examine the effects of two disentanglement-related losses and one newly-introduced generator adversarial loss.

Table 2 shows the results. In the table, $+E_S$ means our model without three kinds of the losses which is equivalent to ACME plus the shape feature encoder, E_S. Because $+E_S$ achieved the higher accuracy than the ACME which is the state-of-the-art on 10k samples, adding the shape feature branch can be regarded as being effective to remove non-recipe information, which might be harmful for cross-modal search, from visual embeddings. The accuracy was improved more by adding either $\mathcal{L}_{G_{img2img}}$ or $\mathcal{L}_{Disen}$. By using all the three losses, the accuracy was greatly improved, and the proposed method outperformed all the current state-of-the-arts greatly. From these results, all the three losses newly proposed in the paper are effective for improving the cross-modal search performance.

In addition, we also evaluated the model trained from scratch in only the second stage by skipping the first stage. The bottom row of the table shows the results, which means that training of the model failed and two-stage training is required. This is because the representation ability of the 8x8x1024 shape map feature is too strong compared to a 128-d visual recipe embedding, and recipe information in addition to non-recipe information is expected to pass through the shape feature branch.

Note that there is no difference regarding retrieval tasks between the only first stage and the two stages, since the encoder of visual embedding, E_V, is frozen at the time of training in the second stage.

4.3 Evaluation on Food Image Synthesis

Since food image synthesis is a by-product of the disentangled recipe embeddings, we evaluated the quality of the synthesized images only for examining the effects of introducing feature disentanglement and two stage training with Fréchet Inception Distance (FID) [9]. In the experiment, we compare FID scores among no disentanglement, only first stage and two-stage training. As the results without disentanglement, we used the ones by ACME [25].

Table 2: Ablation studies on retrieval performance (10k).

	Image-to-Text			Text-to-Image		
Methods	MedR↓	R@1↑	R@5↑	MedR↓	R@1↑	R@5↑
ACME [25]	6.7	22.9	46.8	6.0	24.4	47.9
$+E_S$	6.4	28.1	47.7	5.9	30.1	49.7
$+E_S+\mathcal{L}_{G_{img2img}}$	5.8	29.5	50.0	5.0	31.2	51.4
$+E_S+\mathcal{L}_{Disen}$	5.5	29.7	49.8	5.0	32.2	51.8
ALL	**3.5**	**36.0**	**56.1**	**3.0**	**38.2**	**57.7**
only 2nd stage	361.6	2.7	7.9	288.4	3.6	9.9

Table 3: Comparison on image quality by the FID scores.

Methods	Image2Image ↓	Text2Image ↓
no disentangle (ACME)	183.8	182.9
only first stage (ours)	162.8	168.2
two stages (ours)	**158.9**	**158.6**

Table 3 shows the results, which indicated that introducing feature disentanglement improved the quality of the generated images regarding both image generation from visual embeddings (Image2Image) and image generation from textual embeddings (Text2Image). In addition, the results by two stages were better than the ones by only the first stage, which proved the effectiveness of the proposed two-stage training procedure.

Figure 4 shows some examples of synthesized images by ACME [25], and our two methods. The reconstructed images of the proposed method are closer to the input images than the reconstructed image of ACME. Especially, since the representation ability of the shape map features is strong, almost the identical images are generated.

4.4 Gradual Change of Synthesized Images

In our method, the generated image can be changed by keeping a recipe embedding and changing only a shape vector, and vice versa. To confirm this, we performed morphing on either recipe or shape. No other methods on recipe image generation have had this characteristic so far.

Figure 5 shows the results by gradually changing either a recipe embedding or a shape feature. In case of (a) changing only recipe, while the shape of the images remains unchanged, the features related to the ingredients of the foods, such as color and texture, are changed. In the top row, during the change from salad to plum

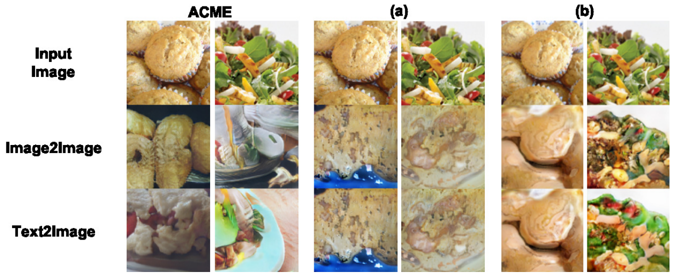

Figure 4: Comparison of the synthesized images with no disentanglement (ACME), (a) only first stage (ours), and (b) two stages (ours).

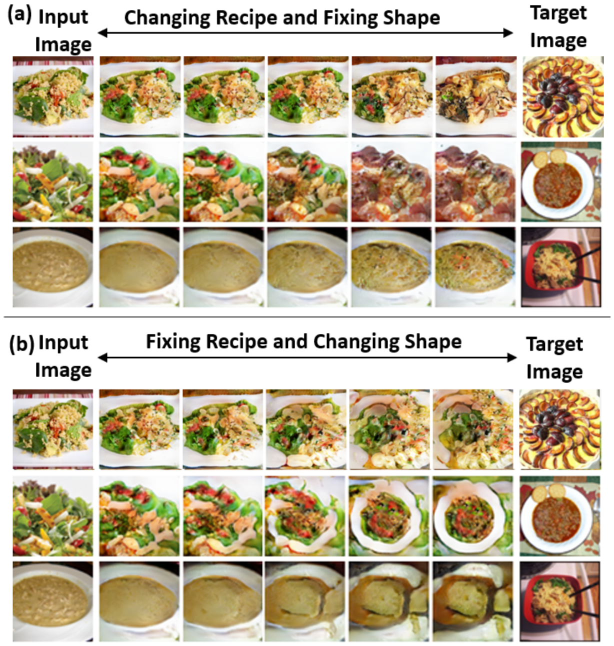
(a) Input Image
Changing Recipe and Fixing Shape
Target Image
(b) Input Image
Fixing Recipe and Changing Shape
Target Image

Figure 5: (a) Synthesized dish images by changing the visual recipe embeddings extracted from two recipe images gradually. (b) Images generated by changing the dish shape vectors. More results are shown at the supplementary material.

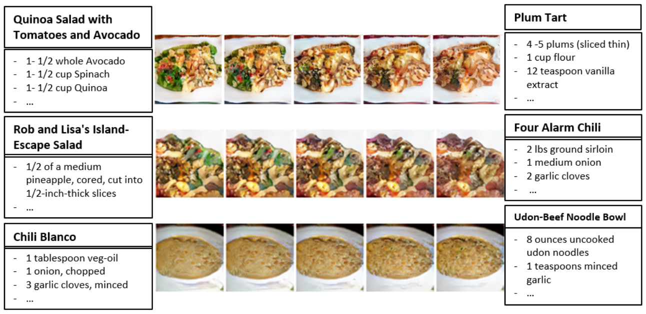
Quinoa Salad with Tomatoes and Avocado
- 1- 1/2 whole Avocado
- 1- 1/2 cup Spinach
- 1- 1/2 cup Quinoa
- ...
Rob and Lisa's Island-Escape Salad
- 1/2 of a medium pineapple, cored, cut into 1/2-inch-thick slices
- ...
Chili Blanco
- 1 tablespoon veg-oil
- 1 onion, chopped
- 3 garlic cloves, minced
- ...
Plum Tart
- 4 -5 plums (sliced thin)
- 1 cup flour
- 12 teaspoon vanilla extract
- ...
Four Alarm Chili
- 2 lbs ground sirloin
- 1 medium onion
- 2 garlic cloves
- ...
Udon-Beef Noodle Bowl
- 8 ounces uncooked udon noodles
- 1 teaspoons minced garlic
- ...

Figure 6: Synthesized dish images by changing the textual recipe embedding between two recipe texts gradually. The recipe images corresponding to each of the recipe texts are shown in the leftmost and rightmost columns in Table 5.

tart, the tart-like color emerged with the shape unchanged. In the second row from salad to chili soup, the green leaves changed to red chili. In the third row from soup to noodles, the texture of noodles were gradually added to the soup.

In case of (b) changing only shapes, the background and non-recipe information which are unrelated to ingredients and cooking procedure, such as plates and serving styles, are changed. In the top row from salad to tart, it still looks like a salad, although the shape of the salad became close to the shape of the plum tart. In the second row from salad to tomato soup, the color of the leaves, which looks like a salad, is kept, while the appearance of a salad is gradually changed from a close-up shot to the overhead shot in which a salad is served in the plate. In the third row from soup to noodles, the shape of the soup bowl is changed from a circle to a square. More results are shown in the supplementary material.

Since RDE-GAN can generate images not only from visual embeddings but also from textual embeddings, we also performed morphing between two textual embeddings to examine the continuity of the textual recipe embedding space. Figure 6 shows the results by changing the textual embedding between the left recipe and the right recipe gradually in each row. In the experiment, we used the recipe texts corresponding to the recipe images shown in Figure 5(a) and the same dish shape features as well. As results, we can see almost the same results in Figure 6 as the results in Figure 5(a), since the visual embedding and the textual embedding within the same recipe are expected to be very close to each other. In the top row the color of salad was changed to tart-like color, in the second row the color was changed to be like chill soup, and in the bottom row the fine-grained noodle textures emerged. In all the cases, the shape remained unchanged. These results indicate that visual and textual embeddings in our embedding space can be exchanged to each other for recipe image synthesis, which means that food image translation by changing recipe texts is possible with our model.

5 CONCLUSIONS

In this paper, we proposed Recipe Embedding and Disentangling GAN (RDE-GAN) which disentangled recipe information from non-recipe information for encoding recipe images. By introducing feature disentangling, we successfully obtained cross-modal recipe embeddings which reflected only the recipe content information by excluding non-recipe information such as dish shape and serving style, and outperformed the state-of-the-art results of the existing baselines on both image-to-recipe and recipe-to-image retrieval with 10k samples. In addition, we synthesized food images from either textual embeddings or visual embeddings by integrating them with dish shape vectors, and showed that we could controlled dish image synthesis in various ways by combining recipe embeddings and dish shape features. Since we can integrate a textual embedding with a dish shape feature, we can change the food category and the ingredients of a given dish image with a recipe text. This is one of the advantages brought only by the integration of cross-modal recipe embedding and disentangled image synthesis.

For future work, we plan to improve both the quality and the size of synthesized food images keeping the performance on cross-modal recipe search. We also plan to apply cross-modal disentangled embeddings into other domains than food recipes.

REFERENCES

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

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. [n.d.]. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc.of the Conference of the North American Chapter of the Association for Computational Linguistics*.

[3] Han Fu, Rui Wu, Chenghao Liu, and Jianling Sun. 2020. MCEN: Bridging Cross-Modal Gap between Cooking Recipes and Dish Images with Latent Variable Model. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*.

[4] I. Goodfellow, A. J. Pouget, M. Mirza, B. Xu, F. D. Warde, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. 2017. Improved training of Wasserstein Gans. In *Advances in Neural Information Processing Systems*. 5767–5777.

[6] Fangda Han, Ricardo Guerrero, and Vladimir Pavlovic. 2020. CookGAN: Meal Image Synthesis from Ingredients. In *Proc. of IEEE/CFV Winter Conference on Applications of Computer Vision*.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*. 770–778.

[8] A. Hermans, L. Beyer, and B. Leibe. 2017. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737* (2017).

[9] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*.

[10] Daichi Horita, Wataru Shimoda, and Keiji Yanai. 2019. Unseen Food Creation by Mixing Existing Food Images with Conditional StyleGAN. In *Proc. of ACMMM Workshop on Multimedia Assisted Dietary Management (MADIMA)*.

[11] X. Huang and S. Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. of IEEE/CVF International Conference on Computer Vision*. 1501–1510.

[12] X. Huang, M. Liu, S. Belongie, and J. Kautz. 2018. Multimodal Unsupervised Image-to-Image Translation. In *Proc. of European Conference on Computer Vision*. 172–189.

[13] Yoshifumi Ito, Wataru Shimoda, and Keiji Yanai. 2018. Food Image Generation using A Large Amount of Food Images with Conditional GAN: RamenGAN and RecipeGAN. In *Proc. of International Workshop on Multimedia Assisted Dietary Management (MADIMA)*.

[14] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*.

[15] H. Lee, H. Tseng, J. Huang, M. Singh, and M. Yang. 2018. Diverse Image-to-Image Translation via Disentangled Representations. In *Proc. of European Conference on Computer Vision*. 35–51.

[16] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. S. Paul. 2017. Least squares generative adversarial networks. In *Proc. of IEEE/CVF International Conference on Computer Vision*. 2794–2802.

[17] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba. 2019. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[18] M. Mirza and S. Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv:1411.1784* (2014).

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *Proc. of International Conference on Learning Representations*. https://openreview.net/forum?id=B1QRgziT-

[20] Siyuan Pan, Ling Dai, Xuhong Hou, Huating Li, and Bin Sheng. 2020. ChefGAN: Food Image Generation from Recipes. In *Proc. of ACM International Conference Multimedia*. 4244–4252.

[21] Yuxin Peng, Jinwei Qi, and Yuxin Yuan. 2017. CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning. In *arXiv:/1710.05106*.

[22] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. 2017. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*.

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*. 815–823.

[24] Bokun Wang, Yang Yang, Xing Xu, Alan Hanjalic, and Heng Tao Shen. 2017. Adversarial cross-modal retrieval. In *Proc. of ACM International Conference Multimedia*. 154–162.

[25] H. Wang, D. Sahoo, C. Liu, E. Lim, and S. CH, Hoi. 2019. Learning Cross-Modal Embeddings with Adversarial Networks for Cooking Recipes and Food Images. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*.

[26] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*. 1386–1393.

[27] Zichen Zan, Lin Li, Jianquan Liu, and Dong Zhou. 2020. Sentence-Based and Noise-Robust Cross-Modal Retrieval on Cooking Recipes and Food Images. In *Proc. of the International Conference on Multimedia Retrieval*. 117–125.

[28] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. 2017. StackGAN: Text to Photo-Realistic Image Synthesis With Stacked Generative Adversarial Networks. In *Proc. of IEEE/CVF International Conference on Computer Vision*.

[29] Bin Zhu and Chong-Wah Ngo. 2020. CookGAN: Causality Based Text-to-Image Synthesis. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*.

[30] Bin Zhu, Chong-Wah Ngo, Jing-jing Chen, and Yanbin Hao. 2019. R2GAN: Cross-Modal Recipe Retrieval With Generative Adversarial Network. In *Proc. of IEEE/CVF Computer Vision and Pattern Recognition*. 11477–11486.