# Cross Modal Respresentation Learning

Shouzhi Fang
Department of Computer Science
Rutgers University
sf716@rutgers.edu

Parth Hasmukh Jain
Department of Computer Science
Rutgers University
pj269@rutgers.edu

Tingcong Jiang
Department of ECE
Rutgers University
tj215@scarletmail.rutgers.edu

Fionna Zhang
Department of Computer Science
Rutgers University
fwz2@scarletmail.rutgers.edu

## Abstract

*Cross-modal recipe retrieval has recently gained a lot of interest due to the importance of food in people's lives and the availability of vast quantities of digital culinary recipes and food images to train machine learning models. Learning a common representation from many perspectives is referred to as cross-modal representation learning. Cross-modal representation learning is important for Retrieval, Translation, and correspondence. In this step, we investigate whether fine-grain representation and correspondence between those entities within images and text improve both the retrieval performance as well as the interpretability of the results. In correspondence we are given a feature in text view we find corresponding feature in image view and vice versa. This project explores correspondence by demonstrating a two streamlined end-to-end model based on well-known and high-performing text encoder transformer and an image processing vision transformer. The Recipe1M [5] dataset, which contains text recipes and related images, is used to investigate the challenge of cross-modal retrieval in this theme. To do so, we conduct a thorough study and ablation research to back up our design decisions.*

***Keywords***:*cross-modal representation,Text-Image Retrieval, Text-mage correspondence*

## 1. Introduction

Given its connection to health, culture, food is one of the most important factors for people. Designing sophisticated tools to explore enormous volumes of data, such as Recipe1M, can assist people in their cooking activities and improve their eating experience, making it an appealing research subject. The availability of such large-scale food datasets has paved the way for new food computing applications, one of the most popular of which is cross-modal recipe retrieval, which aims to create models capable of finding relevant cooking recipes based on food images. To understand cross-modal retrieval we first understand cross-modal representation learning. Cross-modal representation learning seeks to acquire latent semantic representations for a variety of modalities such as texts and photos.

Cross-modal retrieval [3] aims to extract meaningful instances from a different modality, such as an image using text. The fundamental issue is the media gap, which occurs when features from distinct modalities are incompatible, making it difficult to compare them. Since we are working with images and text, we must develop models that combine natural language processing and computer vision to tackle this problem. In the previous step of the project we explored classical multi-view representation learning. We used ResNet [4] features for our image input and BERT features for our text input, then learned common representations using linear models for paired data, using Canonical Correlation Analysis [10, 6], Deep CCA [9], and MLP with Triplet loss [7]. However, the aim in this step is to investigate whether fine-grain representation and correspondence between entities within images and text improve both the retrieval performance as well as the interpretability of the results. Fine grain representation means that given the word "carrot" in the text view, a carrot should also be visible somewhere in the image output. To solve this problem of fine grain representation and correspondences between entities within image and text and to improve the retrieval performance and the interpretability of results, we try

a creative model which makes use of transformer decoder to build relation between image and fine grain text. In this paper we propose two model architecture. Model 1 shares the principle behind the Siamese Network. By controlling the distance between the generated embeddings based on the inputs' labels, we try to maximise the difference if the inputs are from two different classes, and vice-versa if the inputs are from the same class. However, Model 1 does not address the problem of fine grain representation and correspondence. So we additionally propose a second model in which we use multi-head attention to jointly attend to information from the different representation sub spaces of text and image.

## 2. Dataset

We used the Recipe 1 Million (R1M) dataset for testing our models. The dataset consists of 1M text recipes that contain titles,instructions and ingredients in English. Additionally, a subset of 0.5M recipes contain at least one image per recipe. The data is split into 238999 train, 51119 validation and 51303 test image-recipe pairs, in accordance to the official data release provided in R1M.

## 3. Previous Work

### 3.1. CCA

The task of identifying two sets of basis vectors, one for x and another for y, such that the correlations between the projections of the variables onto these basis vectors are mutually maximized is known as canonical correlation analysis. CCA (Canonical Correlation Analysis) is a well-known method for determining the relationships between two sets of multidimensional data. It takes two views of the same collection of items and projects them onto a lower-dimensional realm where their correlation is maximized. CCA has been effectively used in a variety of applications. Table 1 contains our results for both of the retrieval tasks when tested on 1000 samples each. For our ablation studies we found that instructions perform better than ingredients and the average embeddings of all the features gives us the optimum results.

### 3.2. DCCA

Deep CCA (DCCA) uses the idea of Pseudo Siamese Networks (PSN). The only difference between a PSN and a SN is that the two models in the PSN do not share the same parameters. Thus, it is called Pseudo SN. A common use of PSN or SN is to train feature extractors as the objective functions of PSNs are typically distance-related. Therefore, by minimizing the distance between samples with the same label while maximizing the distance between samples different labels, the feature extractors tends to output similar embeddings for the inputs from the same class while output dissimilar embeddings for inputs from different classes. In the case of Deep CCA, it utilizes a similar principle. Instead of using distance-based objective functions, Deep CCA uses an objective function that maximize the correlation between the outputs from a PSN. Given the two inputs ($X_1$,$X_2$), the objective function is defined as following:

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1,\theta_2)}{\operatorname{argmax}} \operatorname{corr}\left(f_1\left(X_1;\theta_1\right), f_2\left(X_2;\theta_2\right)\right) \quad (1)$$

where the $f1$ and $f2$ are the two models in Deep CCA, and $\theta_1$ and $\theta_2$ are the parameters or weights of the two models. By maximizing the correlation between the outputs from Deep CCA, it is intuitive to think that the outputs are in the same share feature space, which is the ultimate goal of a linear CCA. Thus, such an objective function with the structure of PSNs forms the Deep CCA.

### 3.3. MLP with Triplet Loss

Multilayer Perceptrons (MLPs) are deep neural networks that have multiple hidden layers. In general, different objective functions defines the tasks for MLPs, and triplet loss is one of them. Triplet loss is a distance based loss, and it has three inputs: anchor, positive, and negative samples. An anchor sample is a reference sample for the positive and negative samples. Then, the positive sample is the sample that has the same label as the anchor sample has, and the negative sample is the sample that has a different label than the anchor sample has. Next, the triplet loss tries to minimize the distance between the anchor and the positive samples and maximize the distance between the anchor and the negative samples, which is similar to what we have discussed before. Mathematically, triplet loss is as following:

$$L(a,p,n) = \max\left\{d\left(a_i,p_i\right) - d\left(a_i,n_i\right) + \text{margin}, 0\right\}$$

where $a_i$ is the $i_{th}$ anchor sample, $p_i$ is the $i_{th}$ positive sample, and $n_i$ is the $i_{th}$ negative sample. $d(a,b)$ is a distance metric that measure the distance between the two samples $a$ and $b$. The margin is a user defined real value. The neural network will update its parameters during back-propagation only when the distance between the positive pair and the negative pair exceeds the margin. Since triplet loss is a distance-based objective function and our goal is map two inputs into a share feature space, it is intuitive to use a PSN. Therefore, in our proposed work, we use the structure of a PSN that consists of two MLPs. One MLP is used to extract common features from the image feature vectors while another is used to extract common features from the text feature vectors.
The results in Table 2 tell us that the MLP model performs better than Deep CCA when they have the same number of

Table 1: Retrieval metrics for CCA

| | Recipe2Image | | | Image2Recipe | | |
|---|---|---|---|---|---|---|
| | Average Embeddings Of text | Ingredients only | Instructions only | Average Embeddings Of text | Ingredients only | Instructions only |
| MedR | 1.00 | 28.00 | 4.00 | 1.00 | 22.55 | 4.00 |
| R@1 | 0.56 | 0.1089 | 0.2616 | 0.521 | 0.1089 | 0.2512 |
| R@5 | 0.756 | 0.2697 | 0.5697 | 0.732 | 0.2882 | 0.5668 |
| R@10 | 0.81 | 0.3474 | 0.7058 | 0.810 | 0.3893 | 0.6997 |

Table 2: Retrieval metrics for Deep CCA and MLP with Triplet Loss

| Model | Deep CCA | | MLP with triplet Loss | |
|---|---|---|---|---|
| Retrieval | Recipe2Image | Image2 Recipe | Recipe2Image | Image2Recipe |
| MedR | 2.393 | 2.29 | 1.55 | 1.4 |
| R@1 | 0.429 | 0.427 | 0.499 | 0.504 |
| R@5 | 0.744 | 0.745 | 0.772 | 0.787 |
| R@10 | 0.836 | 0.837 | 0.846 | 0.860 |

latent dimensions and the same layer size. There are 2 reasons. Firstly, the MLP model converges faster than Deep CCA while under the same conditions. Secondly, the MLP with triplet loss model takes into consideration the negative sample, and Cosine Similarity loss is more consistent with our target than Deep CCA.

## 4. Cross-Modal Representation Learning Algorithm

In this step, we investigate whether fine-grain representation (e.g., objects/regions in images and words/groups of words in text) and correspondences between entities within image and text improve both the retrieval performance as well as the interpretability of results. We propose 2 attention models.

### 4.1. Model 1

Model 1 as seen in figure 2 shares the principle behind the Siamese Network. It is similar to the model in step 2 of the project where we used an MLP with triplet loss. The differences between Model 1 and the MLP model with triplet loss are that we use attention instead of using a Multilayer Perceptron and the image embeddings are taken from VIT in Model 1 instead of ResNet50. By controlling the distance between the generated embeddings based on the inputs' labels, we try to maximise the difference if the inputs are from two different classes, and vice-versa if the inputs are from the same class. Additionally, the encoders generate self-attention of the embeddings output by the ViT [2] and BERT [1].

The Vision Transformer(ViT) (figure 1) [2] based encoder is used to to learn a mapping function which projects

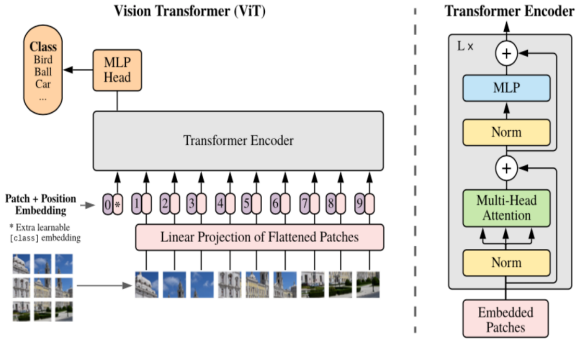



Figure 1: ViT

the input image into the joint image-recipe embedding space. Intuitively speaking, the distance is correlated to the self-attentions. In Model 1, we pass the image input into a Vision Transformer and the text input into BERT. In the Vision Transformer, the input image is split into fixed size patches. Each of these patches are then linearly embedded. We add position embeddings and feed the result to a standard transformer encoder. The transformer encoder from the Vision transformer gives us the image embeddings.

The text input in Model 1 is given to BERT [1], a Pre-trained Deep Bidirectional Transformer used generally for Language understanding. BERT employs a Transformer, an attention mechanism that learns contextual relationships between words (or sub-words) in a text. Transformers incorporate two different mechanisms in their basic form: an encoder that reads the text input and a decoder that gen-

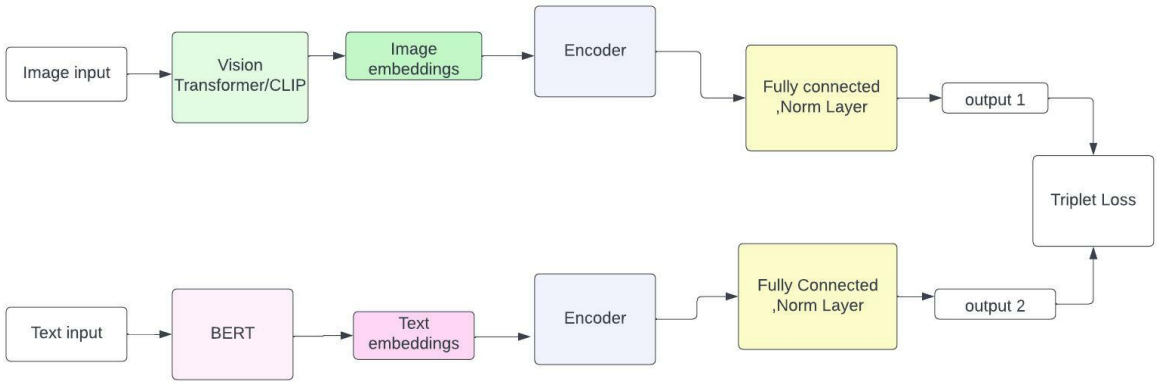



Figure 2: Model 1

Table 3: model1 ablation study result

| | recipe2image | | | | image2recipe | | | |
|---|---|---|---|---|---|---|---|---|
| | medR | Recall@1 | Recall@5 | Recall@10 | medR | Recall@1 | Recall@5 | Recall@10 |
| all_text | 1 | 0.6081 | 0.8489 | 0.9079 | 1 | 0.611 | 0.8498 | 0.909 |
| title | 8.6 | 0.1771 | 0.4197 | 0.5223 | 8.2 | 0.1891 | 0.4317 | 0.5384 |
| ingredient | 2.75 | 0.3545 | 0.6199 | 0.7618 | 2.61 | 0.3674 | 0.6217 | 0.7623 |
| instruction | 2.63 | 0.3672 | 0.6711 | 0.7883 | 2.7 | 0.3592 | 0.6678 | 0.7781 |

erates a job prediction. Only the encoder technique is required because BERT's purpose is to construct a language model. The Transformer encoder reads the entire sequence of words at once, unlike directional models that read the text input sequentially. This feature enables the model to learn the context of a word by looking at its surroundings. In our model, we directly use the embedding features of step 1 and step 2 as the known features and ground truth.

The outputs of the BERT and the ViT are then passed to separate encoders which both employ self-attention [8]. The self-attention mechanism allows the inputs to interact with each other ("self") and find out who they should pay more attention to ("attention"). The outputs are aggregates of these interactions and attention scores. Then, the outputs from the encoders are passed through separate simple fully connected norm layers. Those outputs are given to a triplet loss model which maps two inputs into a shared feature space.

#### 4.1.1 Results

The results can be see in Table 3. From the results we can observe that we get an median rank of 1.0 which is on par with the median ranks from the CCA, DCCA, and MLP with triplet loss models. For the recall rate, we can see that we get generally better results from Model 1 than from CCA, DCCA, or even the MLP model with triplet loss. While looking at the results we notice that the instructions contribute the most but ingredients are not far behind, which was not the case with the previously employed models. To get the optimal results we can use the all text features, which is a concatenation of all the text features.

### 4.2. Model 2

To find whether fine grain representation and correspondences between entities within image and text improve the retrieval performance and the interpretability of results, we propose a creative model which makes use of transformer decoder to build relation between image and fine-grain text. The model structure is in figure 3

- First we concatenate the title, ingredients and instructions. We then we use BERT to extract text features as recipe embeddings.
- We then used BERT to extract fine-grain features from the text data. The output dimension of the BERT encoder is 768. This is then fed into the Key(K) and Value(V) inputs of the decoder.
- Then a ViT is implemented to extract image features to feed into the Query(Q) of the Transformer Decoder.

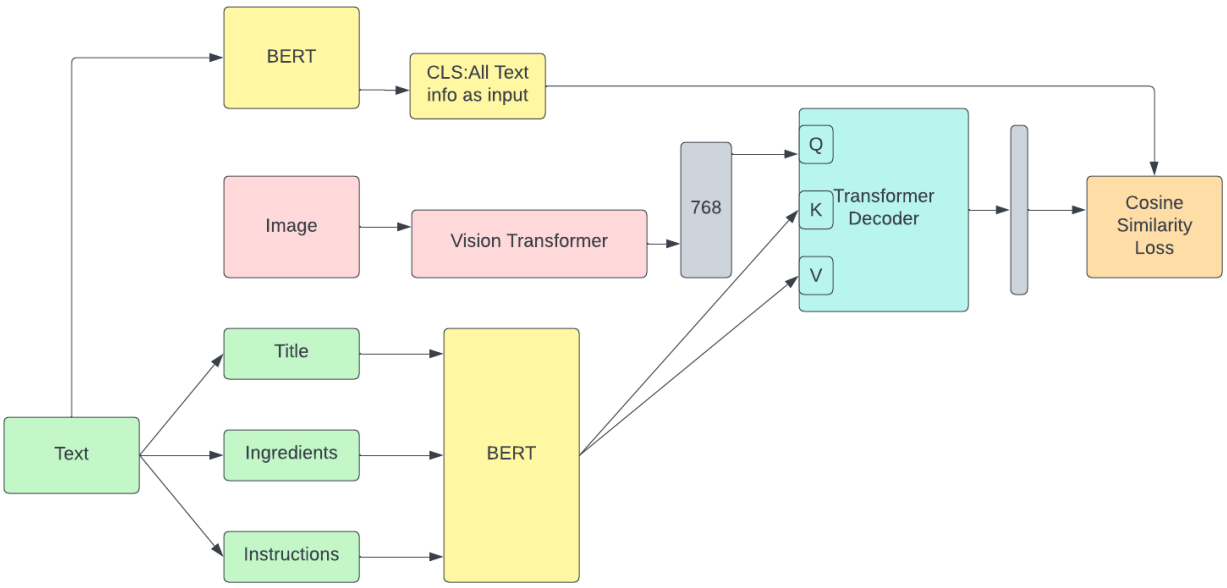



Figure 3: Transformer Decoder to learn the correspondence between fine grain text feature and image feature

Table 4: Model2 best medR and Recall result

| Sample result for 4 head ,2 layer | medR | Recall@1 | Recall@5 | Recall@10 |
|---|---|---|---|---|
| Model 2 | 50 | 0.0803 | 0.1107 | 0.1638 |

- We use multi-head attention to jointly attend to information from the different fine-grain representations of text and image.

- For evaluation, we use cosine similarity loss on the output of the Transformer decoder with recipe text embedding from BERT.

The creativity of Model 2 is the utility of the Transformer Decoder. From the properties of Attention Architecture, we know:

$$Q = Linear_q(X) = XW_q$$
$$K = Linear_k(X) = XW_k$$
$$V = Linear_v(X) = XW_v$$
$$X_{attention} = Self - Attention(Q, K, V)$$
$$Weight\ Matrix = SoftMax(\frac{QK^T}{\sqrt{d_k}})$$

Theoretically, the Weight Matrix represents the weights of each input image corresponding to each fine grained text features. The result is showing in Table 4. However, Model 2 converges on $medR = 50$.

### 4.3. Fine-Grain Features Qualitative Analysis

In this subsection, we perform a qualitative analysis on fine-grain features. Fine-grain features are sub-features that represent only a portion of the global input. For instance, a fine-grain feature of a sentence is a word or set of words. Since fine-grain features are sub-features, we cannot only use global feature extractors such as ResNets to extract them. To address this challenge, we utilize transformers. Transformers are feature extractors that output the feature embeddings and their corresponding importance, which is known as attention. This helps for importance, since we can see the relationship between the sub-features in a given text and the sub-features in a given image. By doing so, we have a better understanding of how feature extractors comprehend the elements in the given inputs. To link the attentions of text features to image features, we first rank the fine-grain features by their attentions in descending order. Then, we match the fine-grain features from both models by their ranks. The following is a example case of image to text retrieval:

**Example Case:**

In this case, the choose the top 3 results from the re-

trieval results and demonstrate the how fine-grain features are linked together as shown in Fig.4. First of all, all 3 results show that both the text and image feature extractors put the most importance on the type of the food. For instance, the most confident result shows that the text "Muffins" is linked with the sub-regions in the query image where cookies exist. Although this is a mismatch, the result indeed shows that it approaches the image to text task in human-like ways. Furthermore, the cookies in the query image do have textures that are similar to muffins' textures. Secondly, the feature extractors further show their extreme attention paid on the text or sub-images that are related to the types of foods as they connected the word "chips" with the sub-regions where cookies exist. Since they are mismatched, this is a sign of lacking of training of our models. Because training attention models are complex and time-consuming, we do not have enough time to perfect our models.

We can observe similar trends in the second most confident result, which is a positive match. First of all, the feature extractors links the text "Cookies" in the title with sub-regions in which cookies are present, which further confirms our assumption that the feature extractors are approaching the retrieval tasks in human-like ways. Secondly, both feature extractors link the text description of textures with the textures in images. For instance, the text "375 degrees" is linked with the texture of cookies as the texture of cookies is the result of baking in an oven.

Although there are mismatching features, our models show that fine-grain cross-modal representation learning is a prominent approach. Furthermore, without supervision, the features extractors show that they are approaching the representation learning problems in human-like behaviors. Last but not least, with more training iteration and hyper-parameter fine-tuning, we can significantly improve the recall rates and accuracy.

## 5. Conclusion and Future Work

From our experiment result, Model 1 performs better than previous non-attention model(CCA, DCCA, MLP with Triplet loss). However, it is hard to explain which of the fine grain features has more contribution to our cross-modal representation.

In our design of Model 2, ideally given an image, it would find the identify the most related fine grain text information with highest weight from the multi-head attention architecture in the following form:

$$SoftMax(\frac{QK^T}{\sqrt{d_k}})$$

However, Model 2 is hard to converge. It is hard for Model 2 to find the fine-grain representation improving the performance of results. In our opinion, the main reason is that Bert poorly extracts features of long sentence(more than 512 tokens), which exceeds the model's limit.

To solve this problem, we should focus on the extraction of features from long sentences. Additionally, if we could find more fine-grain labeled dataset, we can try model 3 5. We can fine-tune our Bert and ViT model, and train self-attention models from scratch. In order to improve the interpretability of our model, we will use a cross-entropy loss function on a labeled dataset.

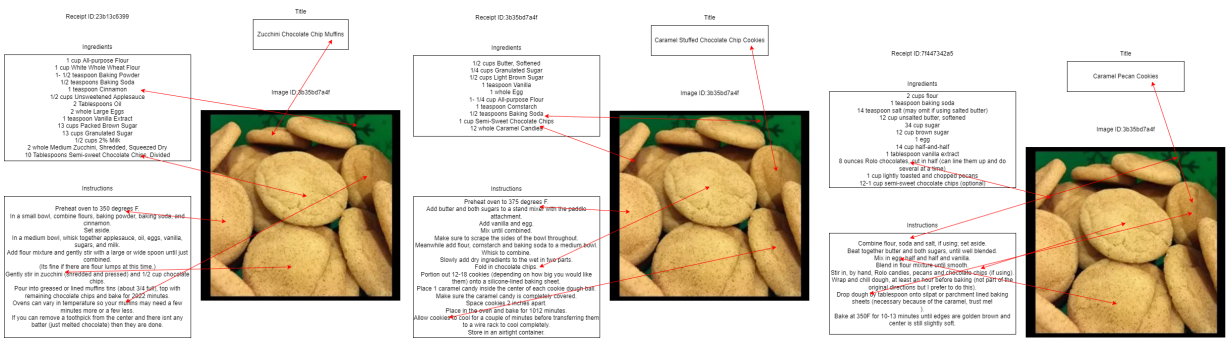
## 6. Contributions

**Shouzhi Fang (sf716)**: model design, code implementation and section 5, 6 in final report.

**Tingcong Jiang (tj215)**: literature reviews, fine-grain features qualitative analysis.

**Parth Jain (pj269)**:litreture review,Section 1,2,3 and 4,Model evaluation-ablation studies

**Fionna Zhang (fwz2)**: general report writing



(a) (b) (c)

Figure 4: a): the most confident text to image retrieval result. b): the second most confident text to image retrieval result. c): the third most confident text to image retrieval result. The red arrow lines indicate the correspondence of fine-grain features.

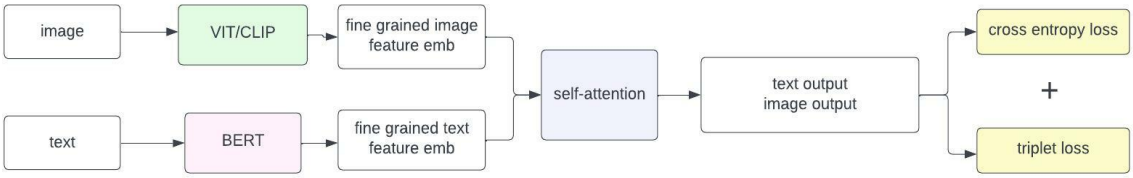



Figure 5: fine-tune encoder and self-attention training from scratch

## References


[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. 3

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 3

[3] R. Guerrero, H. X. Pham, and V. Pavlovic. Cross-modal retrieval and synthesis (x-mrs): Closing the modality gap in shared representation learning, 2020. 1

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. 1

[5] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images, 2018. 1

[6] K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. 1

[7] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. 1

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017. 4

[9] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning: Objectives and optimization, 2016. 1

[10] Wikipedia contributors. Canonical correlation — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Canonical_correlation&oldid=1072473049, 2022. [Online; accessed 8-May-2022]. 1