Project Report

A Comparative Study of Accuracy Driven Model for an Image to Recipe Retrieval Algorithm

Team no.: 11

Team Members: Anmol Sharma (Sharma265) (G)

Karan Vikyath Veeranna Rupashree (veerannarupa) (G)

Siddharth Baskar (sbaskar2) (G)

Date of Submission: 12/15/2022

Abstract

In this project, four different algorithms were used which would give out the ingredients any dish would have when provided with the image of the dish and then each algorithm was compared with the other to determine which one provided the most accurate results. The four algorithms that were implemented only to encode images are ResNet-50, VGG16, ViT, and DenseNet, further, LSTM and transformer were used to train the model in the text aspect i.e., the ingredients of the dish. Furthermore, the value of k is taken as 1 so that, the models, as an outcome, provide the three closest dishes resembling the dish in the input image. The working of this could be compared to the PageRank system where the models rank the dishes according to the similarity of the dishes to the input dish and then the top three dishes are provided in the output.

1) Introduction

Food is one of the most important requirements of the human body. There are many vitamins and nutrients that are needed by the body in appropriate amounts to function in an optimum manner. These nutrients and vitamins are present in different ingredients in varying quantities and these ingredients make up the dishes that humans intake. Hence, it is necessary for health-conscious people to be knowledgeable of the nutrients that they are consuming in a day. This could be easily tracked by knowing the ingredient intake. So, to simplify and ease the process of accessing the information regarding the ingredients that any dish might have, different models have been used in this project to method is proposed to in this project which will give out the ingredients and the recipe of the dish. In this project, VGG16, DenseNet-121, ViT and ResNet-50 models are used to identify a recipe for the dish from the image of that dish. Further, as a future scope, the calorie count could also be provided for each dish and an app could be developed for user interaction which would be linked to the model and alert the user of the calorie count of either one or a combination of different dishes.

While conducting the background check for this project it was discovered that there has been tremendous improvement in the collection of food-recipe datasets during the last decade. Most early research used a traditional recognition process, concentrating on feature combinations and specialized datasets (mostly Asian cuisine). However, in 2014, Bossard et al. took a stride ahead by introducing the Food-101 visual classification dataset, which had 101k images divided equally across 101 classes and had a baseline accuracy of 50.8%.[1] Chen and Ngo presented another enormous dataset with 65k recipes and 110k photos two years later, in 2016, but this time it only covered Chinese food.[2] Salvador et al. published the Recipe1M+ dataset in 2017, which is a new large-scale, structured corpus of over one million culinary recipes and 13 million food photos.[3] To date, this is the largest publicly available collection of food and recipe data, allowing high-capacity models to be trained on aligned, multi-modal data. As a result, we chose this dataset for our research. A variety of measures have also been attempted to address the issue of food recognition. Yang et al. proposed learning spatial correlations between ingredients using paired characteristics in 2010.[4] This, however, was only going to work for uniform meals. Bossard et al. presented a novel method for mining discriminative components using Random Forests in addition to the Food-101 dataset. Random Forests were utilized to discriminatively group superpixels into groups (leaves), and the leaf statistics were then used to build features. The researchers utilized a distinctiveness metric to blend similar leaves before classifying the food photos using a support vector machine (SVM). Herranz et al. suggested an extended multi-modal paradigm a few years later that investigates how visual content, context, and external knowledge might be integrated into food-oriented apps.[5] These were the most relevant studies that have been conducted on the research area that the current project is based.

In this project, the objective was to retrieve the recipe for a given food image using cross modal retrieval method and to compare and determine the text and image encoder that would be best suited for a small dataset. In order to obtain the objective, LSTM and transformer were used for text encoding and image encoders were used to perform cross modal retrieval.

2) Data

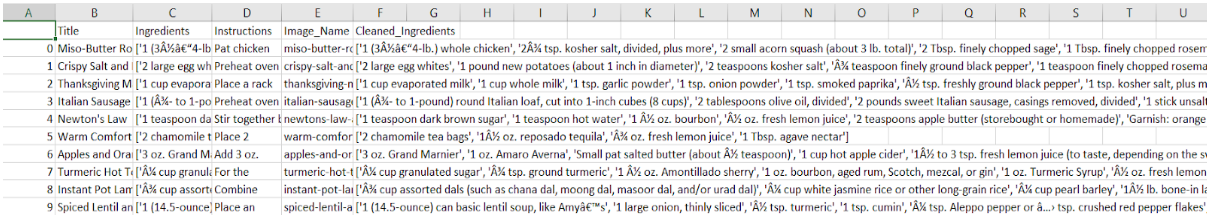
As the food images are very complex and understanding of the features and textures of the food present in an image is very difficult. Due to this reason, various previously available datasets were scraped to get the overall dataset that was used in this project. These include Recipe1M+[6] Dataset, Food101 Dataset[7], and Food Ingredients and Recipes Dataset with Images.[8] The images obtained from these sources totalled up to more than one million. From these one million pools of images, a small subset of 15000 images were used in this project. All these images are of the dimensions 340 x 340 with all three RGB channels.

Pre-processing

Various pre-processing methods like center crop, random horizontal flipping, center crop alone, scaling, and rotation was initially performed, and it was then observed that random crop followed by center crop yielded the best results. The CSV obtained is depicted in Fig 1 and a few of the images used are shown in Fig 2.

Training

From this subset of 15000 images, 80% of the images were used for training, 10% for testing, and 10% for validation. Furthermore, for training a batch size of 30 images were chosen.



A	B	C	D	E	F
	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients
0	Miso-Butter Ro	['1 (3Â½â€“4-lb	Pat chicken	miso-butter-r	['1 (3Â½â€“4-lb.) whole chicken', '2Â¾ tsp. kosher salt, divided, plus more', '2 small acorn squash (about 3 lb. total)', '2 Tbsp. finely chopped sage', '1 Tbsp. finely chopped rosem
1	Crispy Salt and	['2 large egg wh	Preheat oven	crispy-salt-an	['2 large egg whites', '1 pound new potatoes (about 1 inch in diameter)', '2 teaspoons kosher salt', 'Â¾ teaspoon finely ground black pepper', '1 teaspoon finely chopped rosema
2	Thanksgiving M	['1 cup evapora	Place a rack	thanksgiving-n	['1 cup evaporated milk', '1 cup whole milk', '1 tsp. garlic powder', '1 tsp. onion powder', '1 tsp. smoked paprika', 'Â½ tsp. freshly ground black pepper', '1 tsp. kosher salt, plus m
3	Italian Sausage	['1 (Â¾- to 1-po	Preheat oven	italian-sausag	['1 (Â¾- to 1-pound) round Italian loaf, cut into 1-inch cubes (8 cups)', '2 tablespoons olive oil, divided', '2 pounds sweet Italian sausage, casings removed, divided', '1 stick unsalt
4	Newton's Law	['1 teaspoon da	Stir together	newtons-law-	['1 teaspoon dark brown sugar', '1 teaspoon hot water', '1 Â½ oz. bourbon', 'Â½ oz. fresh lemon juice', '2 teaspoons apple butter (storebought or homemade)', 'Garnish: orange
5	Warm Comfort	['2 chamomile t	Place 2	warm-comfor	['2 chamomile tea bags', '1Â½ oz. reposado tequila', 'Â¾ oz. fresh lemon juice', '1 Tbsp. agave nectar']
6	Apples and Ora	['3 oz. Grand M	Add 3 oz.	apples-and-or	['3 oz. Grand Marnier', '1 oz. Amaro Averna', 'Small pat salted butter (about Â½ teaspoon)', '1 cup hot apple cider', '1Â½ to 3 tsp. fresh lemon juice (to taste, depending on the s
7	Turmeric Hot T	['Â¼ cup granul	For the	turmeric-hot-t	['Â¼ cup granulated sugar', 'Â¾ tsp. ground turmeric', '1 Â½ oz. Amontillado sherry', '1 oz. bourbon, aged rum, Scotch, mezcal, or gin', '1 oz. Turmeric Syrup', 'Â½ oz. fresh lemon
8	Instant Pot Lam	['Â¾ cup assort	Combine	instant-pot-la	['Â¾ cup assorted dals (such as chana dal, moong dal, masoor dal, and/or urad dal)', 'Â¼ cup white jasmine rice or other long-grain rice', 'Â¼ cup pearl barley', '1Â½ lb. bone-in la
9	Spiced Lentil an	['1 (14.5-ounce)	Place an	spiced-lentil-a	['1 (14.5-ounce) can basic lentil soup, like Amyâ€™s', '1 large onion, thinly sliced', 'Â½ tsp. turmeric', '1 tsp. cumin', 'Â¼ tsp. Aleppo pepper or â…› tsp. crushed red pepper flakes'

Fig 1: CSV

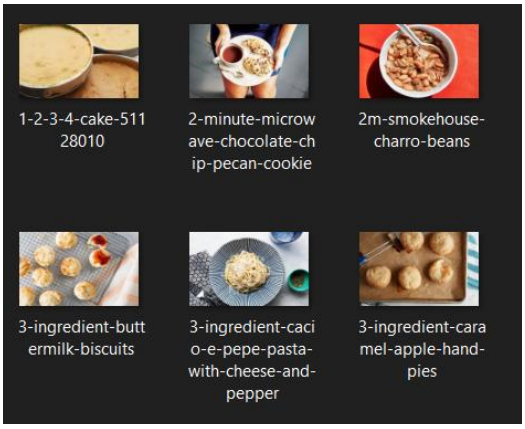

1-2-3-4-cake-51128010
2-minute-microwave-chocolate-chip-pecan-cookie
2m-smokehouse-charro-beans
3-ingredient-buttermilk-biscuits
3-ingredient-cacio-e-pepe-pasta-with-cheese-and-pepper
3-ingredient-caramel-apple-hand-pies

Fig 2: Images from the Dataset

3) Method

The aim of the project is to achieve maximum similarity between the recipe encoding and the matched-up image encoding and to reduce the similarity of non-matched-up pairs of the recipe and image encoding. The methodology followed in this project is depicted in Fig 3 and is further explained in the following subsections.

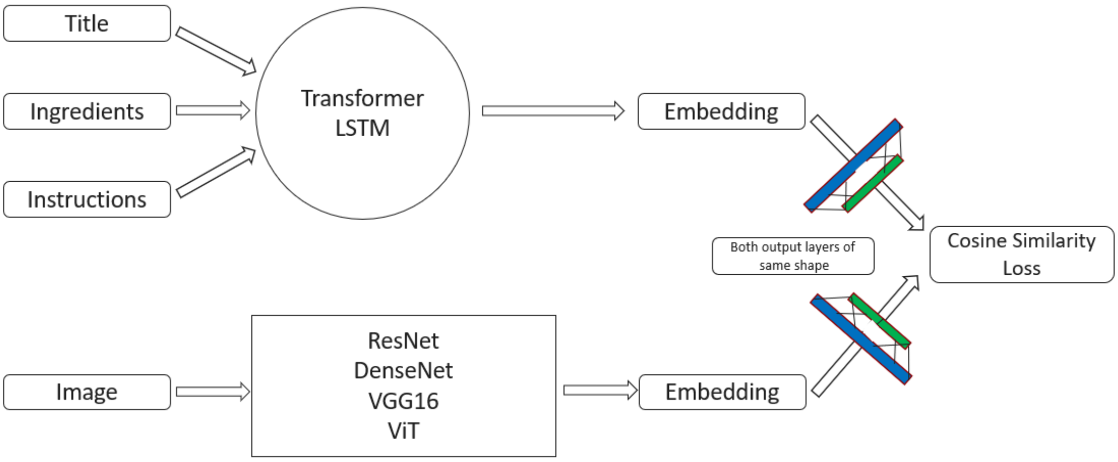


Fig 3: Methodology

3.1) Models

ResNet-50

The ResNet architecture introduces the simple concept of adding an intermediate input to the output of a series of convolution blocks to prevent the Degradation Problem. The reason for using ResNet – 50 was because of its bottleneck design for the building block. The number of parameters and matrix multiplications are reduced because of the 1x1 convolutions in the residual block, which allows faster training of each layer. It uses only 240M FLOPs by reducing the number of rows and columns by a factor of 2 and further reduction in max pooling operation. Due to the lack of concatenation operations, it has low GPU usage as compared to DenseNet.

DenseNet-121

DenseNet (dense convolutional network) is a logical extension of Resnet to a certain degree. The degradation problem is solved by concatenation operations of an intermediate input to the output as opposed to the addition operator in ResNet.

Consequently, the equation reshapes again into:

$$x_{l} = H_{l}([x_{1,} x_{2,\ldots,} x_{l-1})]$$

Due to its improved accuracy in battling vanishing-gradient problem in high-level neural networks, DenseNet was used in this project. It also strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters. Due to its unique approach to implement concatenation operations, the network can be more compact, i.e., the number of channels can be fewer. Furthermore, the errors can also be easily propagated to the preceding layers.

VGG16

VGG 16 instead of having many hyperparameters, focuses on having convolution layers of 3x3 filter with a stride 1 and always uses the same padding and max pool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. It has two fully connected layers preceded by a softmax for the output at the end. The 16 in the name refers to its 16 weighted layers. This network is a large network, and it has about 138 million parameters.

The main reason for the use of VGG16 is that it uses very small receptive fields as opposed to AlexNet. The decision function has three ReLu units instead of one, which makes it more discriminative. Moreover, the VGG model can have a considerable number of weight layers due to the small size of the convolution filters; of course, more layers mean better performance.

VisionTransformer

The visual transformer divides an image into fixed-size patches, correctly embeds each of them, and includes positional embedding as an input to the transformer encoder. Moreover, ViT models outperform CNNs by almost four times when it comes to computational efficiency and accuracy. The self-attention layer in ViT makes it possible to embed information globally across the overall image. The model also learns on training data to encode the relative location of the image patches to reconstruct the structure of the image. The transformer encoder includes: Multi-Head Self Attention Layer (MSP): This layer concatenates all the attention outputs linearly to the right dimensions. The many attention heads help train local and global dependencies in an image. Multi-Layer Perceptrons (MLP) Layer: This layer contains a two-layer with Gaussian Error Linear Unit (GELU). Layer Norm (LN): This is added prior to each block as it does not include any new dependencies between the training images. This thereby helps improve the training time and overall performance. Moreover, residual connections are included after each block as they allow the components to flow through the network directly without passing through non-linear activations.

3.2) Image Encoding

Here, in this project, ResNet-50, VGG16, DenseNet, and Vision Transformer are used to embed the images, and then, transfer learning methods are implemented to accelerate the training process by using CNNs that were pre-trained on ImageNet. These models are incorporated by removing the last softmax classification layer and connecting the rest to the joint embedding model. The flow of these encodings are shown in Fig 3.

3.3) Recipe Encoding

Recipe encoding was performed with two different methods, LSTM and Transformer. In the case of LSTM, a bi-directional LSTM was used for ingredients because of its unordered nature in the data, and a two-stage forward LSTM was used for the recipe instructions. For the transformer method, we encoded each sentence with a 2 - layer transformer network of dimension 512, each with 4 attention heads.

3.4) Joint Encoding

The image encoding and the recipe encoding that is performed in the previous step are then mapped to a joint space of embedding to form the overall recipe encoding which is then used to obtain the performance evaluation of the model.[9] In this, let v_k and r_k be the image and recipe encoding respectively. The connected map is given by the equation:

$$\varphi^R = W^R_{\mathrm{r}_k} + b^R, \varphi^v = W^R_{\mathrm{v}_k} + b^R$$

Where, W^R, W^b, $v^{R,}$ and v^b are also the parameters that have to be learned.

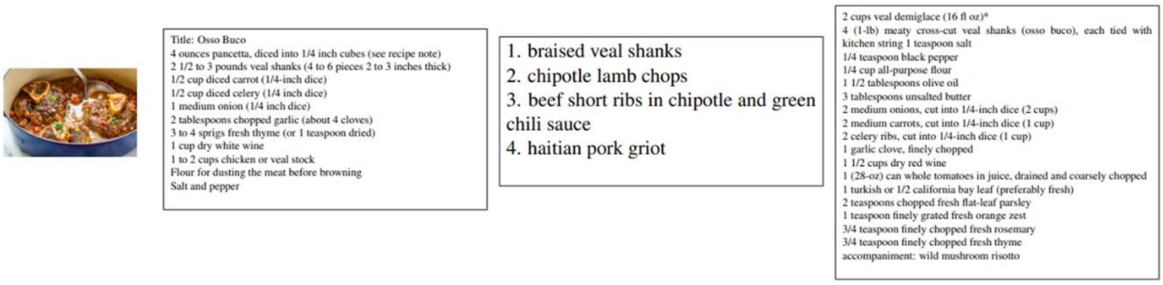
From the image set, with a random probability of 25%, the matching image-recipe pair is selected and with the remaining 75% of probability, the non-matched-up pair is chosen. For the loss function, Cosine similarity loss was used where the model is trained end-to-end with the negative and positive image recipe (v,R) pairs. This function is given by:

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

Positive pair is denoted by y = 1 while the negative pair is done by y = -1 where α is the margin and cos(.) gives the normalized cosine similarity. This loss was further minimized with the help of the ADAM optimizer.

4) Results

Fig 4 depicts the image to recipe retrieval with the DenseNet-121 model. The figure shows that the DenseNet-121 model is able to identify the ingredients that are present in the dish present in the testing image and gives out two text boxes that contain ingredients and instructions. As the value of K is 1, therefore, it gives the ingredients and recipe of the dish which resemble the test image the most.



Title: Osso Buco
4 ounces pancetta, diced into 1/4 inch cubes (see recipe note)
2 1/2 to 3 pounds veal shanks (4 to 6 pieces 2 to 3 inches thick)
1/2 cup diced carrot (1/4-inch dice)
1/2 cup diced celery (1/4 inch dice)
1 medium onion (1/4 inch dice)
2 tablespoons chopped garlic (about 4 cloves)
3 to 4 sprigs fresh thyme (or 1 teaspoon dried)
1 cup dry white wine
1 to 2 cups chicken or veal stock
Flour for dusting the meat before browning
Salt and pepper

1. braised veal shanks
2. chipotle lamb chops
3. beef short ribs in chipotle and green chili sauce
4. haitian pork griot

2 cups veal demiglace (16 fl oz)*
4 (1-lb) meaty cross-cut veal shanks (osso buco), each tied with kitchen string 1 teaspoon salt
1/4 teaspoon black pepper
1/4 cup all-purpose flour
1 1/2 tablespoons olive oil
3 tablespoons unsalted butter
2 medium onions, cut into 1/4-inch dice (2 cups)
2 medium carrots, cut into 1/4-inch dice (1 cup)
2 celery ribs, cut into 1/4-inch dice (1 cup)
1 garlic clove, finely chopped
1 1/2 cups dry red wine
1 (28-oz) can whole tomatoes in juice, drained and coarsely chopped
1 turkish or 1/2 california bay leaf (preferably fresh)
2 teaspoons chopped fresh flat-leaf parsley
1 teaspoon finely grated fresh orange zest
3/4 teaspoon finely chopped fresh rosemary
3/4 teaspoon finely chopped fresh thyme
accompaniment: wild mushroom risotto

Fig 4: Prediction by DenseNet-121

Using the cosine loss the relative accuracy of encoding of images was tested against the other images attached to the same recipe. Here value ‘0’ shows the maximum similarity between the two images. By using this loss, it was quantified on how the found image encoding is for an image and thus calculate the accuracy of the resulting recipe that the model will give.

Table 1 depicts the comparison of performance metrics obtained from all the text encoders and image encoders.

Table 1: Comparison of Text Embeddings

Modals	ResNet-50	ResNet-101	DenseNet-121	ViT
Cosine Loss	0.365	0.281	0.224	0.106

Table 2: Comparison of Different Image Embeddings

Modals	medR	R@1	R@5	R@10
LSTM	12	12.3	31.7	42.2
Transformer	10	15.6	36.2	47.5

Modals	**medR**	**R@1**	**R@5**	**R@10**
ResNet50	16.2	0.09	0.28	0.41
DenseNet-121	15.4	0.109	0.31	0.44
ViT	14.1	0.12	0.32	0.46
VGG16 - Baseline	17.5	0.093	0.24	0.38

Here medR is the median rank among where true positives are returned. Therefore, high performance comes with a lower medR score.

Given a food image R@k calculates the proportion of times that the correct recipe is found within top-k retrieved. The performance is directly proportional to the Recall value.

In text encoding, transformers performed better than LSTM because of their superior performance in NLP tasks. In image encoding Densenet-121 performed better than Resnet-50 because of better information and gradient flow, also it is easy to manage very deep layers in DenseNet. Overall, ViT outperforms the other model because of the self-attention layers which help the network to learn the hierarchies and alignments present inside the image.

5) Discussion

Although good results were achieved with the ViT image encoder and Transformer text encoder still the results obtained in this work are not comparable with the results yielded from the original Recipe1M+ dataset. This is because the dataset used in this project is only around 1/40 of the size of the original dataset. Also, the variance of food images due to the reduced dataset hampers the performance of the model. So it was concluded that to do such a task there is a requirement for a huge training dataset.

Another shortcoming of the model is the inability to train large batch sizes, which is known to lead to fewer parameter updates and greater parallelism. In this project a batch size of 30 was used which is almost half used in the original paper due to GPU constraints.

By resolving many shortcomings from other efforts, the cross-modal retrieval problem in the food domain was examined in this work. At first, a hierarchical Transformer-based textual representation model was provided that outperformed LSTM-based recipe encoders. Secondly, an easy-to-add self-supervised loss to account for relationships between various recipe components was provided.

The retrieval outcomes dramatically over intermediate recipe representations. Additionally, this loss enables us to train on both paired and unpaired recipe data—that is, recipes without images—leading to an even greater improvement in performance. The contributions made in this project helped in producing cutting-edge outcomes in the Recipe1M dataset.

5) References

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

[2] C.-w. N. Jing-jing Chen, “Deep-based ingredient recognition for cooking recipe retrieval,” ACM Multimedia, 2016.

[3] Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber, Ingmar, and Toralba, Antonio. Learning cross-model embeddings for cooking recipes and food images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[4] Yang, S.L., Chen, M., Pomerleau, D., Sukthankar, R.: Food recognition using statistics of pairwise local features. In: CVPR (2010)

[5] L. Herranz, W. Min, and S. Jiang, “Food recognition and recipe analysis: integrating visual content, context and external knowledge,” CoRR, vol. abs/1801.07239, 2018. [Online]. Available: http://arxiv.org/abs/1801.07239

[6] Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., ... & Torralba, A. (2019). Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence, 43*(1), 187-203.

[7] Bossard, L., Guillaumin, M., & Gool, L. V. (2014, September). Food-101–mining discriminative components with random forests. In *European conference on computer vision* (pp. 446-461). Springer, Cham.

[8] Sakshi Goel, Amogh Desai, Tanvi. (2020, August). Food Ingredients and Recipes Dataset with Images, Version 1.

[9] Zhu, K., Sha, H., & Meng, C. ChefNet: Image Captioning and Recipe Matching on Food Image. Training, 720639, 619-508.