International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Image-to-Recipe Translation using Multi-model Architecture

Sudarshan Basawaraj Pune[1], Rupesh Mahal[2], Vishwas M H[3], Shivansh Singhal[4], Mohamadi Ghousiya Kousar[5]

[1,2,3,4]*Student, Department of Computer Science* and Engineering*, Sir M. Visvesvaraya Institute of Technology, VTU, Bengaluru, India*

[5]*Assistant Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, VTU, Bengaluru, India*

Abstract - *Food photography has become a very popular trend in this era. Social media sites are flooded with images of food every day. Many people will not be able to figure out what dish it is, and it will be extremely difficult to determine it accurately. There are many diet-conscious people who also face the same problem. So, there was a need for a system that provides the details of any dish. Hence, we will use image processing to process an input image, classification methods and convolutional neural networks to implement a system that will be able to classify and label the unique ingredients in the picture, and finally, give the recipe of a dish. As the largest freely accessible collection of recipe data, Recipe1M offers the opportunity to train high-capacity models on aligned, multimodal data. Furthermore, we demonstrate that regularization by introducing a high-level classification technique improves retrieval overall performance to rival that of humans and also empowers semantic vector math.*

1. INTRODUCTION

Nutrition is one of the most important factors when it comes to human well-being. Every day, innumerable food pictures are uploaded by users on social networks; from the first home-made cake to the most expensive dish, the joy is shared with you whenever a dish is successfully prepared. Food is fundamental to human existence. Not only does it provide us with energy but it also defines our identity and culture [11, 20]. It is a fact that good food is appreciated by everyone, irrespective of how different people may be from one another.

Food culture has been spreading quite ever within the current digital era, with many of us sharing pictures of food they are eating across social media [22]. Sites like Facebook, Instagram, and twitter are overflowing with pictures and videos of food. In addition, eating patterns and the culture of cooking has evolved over time. In the past, food was mostly prepared at home, but nowadays we frequently consume food prepared by catering and restaurants. Thus, access to detailed information about prepared food is limited, and consequently, it is hard to know precisely what we eat.

The last few years have witnessed outstanding improvements in visual recognition tasks such as natural image classification [11, 22], object detection [16, 18], and semantic segmentation [10, 24]. However, in comparison to natural image understanding, food recognition poses additional challenges, since food and its components have high intraclass variability and present heavy deformations that arise throughout the cooking process.

Hence, we aim to create a system that will help us obtain the ingredients and recipes of the food we eat, using just their images. We are using technologies such as image classification, convolutional neural network, transformer model, recurrent neural networks, and attention strategies for the instruction decoder, and multi-modal embedding, to achieve our goal.

Fig-1: Example of a generated recipe.

2. RELATED WORK

Food Understanding: In past large-scale food dataset were introduced, such as Food-101 [1] and Recipe1M [3] together with a recently help iFood challenge -2019 has enabled significant achievements in visual food recognition, by providing reference benchmarks to train and compare machine learning approaches. As a result, there is currently a vast literature in computer vision dealing with a variety of food-related tasks. Subsequent works tackle more challenging tasks such as estimating the number of calories from a given food image.

Early attempts on food-recipe generation [8] where they compared and evaluated popular text-based and vision-

based technologies on a very large multimodal dataset containing about 101k recipes corresponding to 101k food categories. They proposed a real application for daily users to identify recipes. This application is a web search engine that enables any mobile user to submit a query image and retrieve the most relevant recipes in their dataset.

Significant efforts were made [5] proposed deep architectures for simultaneous learning of ingredient recognition and food categorization, by making use of the mutual relationship between them. They learned semantic labels of ingredients and therefore the deep features are then applied for the retrieval of recipes. A multi-task deep learning model is used to achieve this goal. Recent results on large datasets showed better outcome [3] where they trained a neural network on a huge dataset of over 1 million recipes and 800,000 images to find a joint embedding of recipes and images that produces accurate results on image-recipe retrieval jobs.

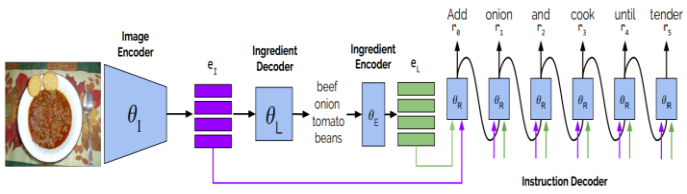

Fig-2: Multi-model architecture

3. Generating recipe from image

It is a challenging task to generate a recipe (title, ingredients, and instructions) from an image. It also demands a contemporaneous understanding of the ingredients composing the dish as well as the transformations they went through, for e.g. slicing, blending, or mixing with other ingredients. Instead of predicting recipes directly from an image, we went for creating a pipeline that benefited from an intermediate step providing the prediction of an ingredient list. Then the sequence of instructions would be generated and it would be conditioned on both images as well as its corresponding list of ingredients. There will be an interplay between the image and ingredients, which provides additional insight and will show later were processed to produce the resulting dish given.

Figure 2 shows detailed information of our approach. Our recipe generation system takes a food image as an input and as an output, a sequence of cooking instructions is generated by means of an instruction decoder. Further, it takes input from two embeddings. The first represents visual features that are extracted from the image and second embedding encodes the ingredients extracted from the image. Subsection 3.1 depicts the introduction of our transformer-based instruction decoder, which permits us to review the transformer. As a result of the study, we are now able to predict ingredients in an order less manner (subsection 3.1). Therefore, reviewing the optimization is shown in subsection 3.2.

3.1 Cooking Instruction Transformer

For a given input image with associated ingredients, we focus to predict a sequence of instruction R = (*r1, ..., rT*) (where *rt* denotes a word in the sequence) by means of an instruction transformer [2]. The first instruction that is predicted is the title of the recipe. The transformer is conditioned jointly on two inputs: the ingredient embedding *eL* and the image representation *eI* and. We used ResNet-150 [7] to extract image representation and to get ingredient embedding *eL* by the usage of a decoder architecture to predict ingredients, followed by a single embedding layer mapping each ingredient into a fixed-size vector. The instruction decoder consists of transformer blocks, each of them containing two attention layers followed by a linear layer [2]. The first attention layer uses a self-attention strategy over previously generated outputs and the second one attends to the model conditioning in order to refine the self-attention output. t. The transformer model is composed of multiple transformer blocks followed by a linear layer and a SoftMax nonlinearity that provides a distribution over recipe words for each time step *t.* Figure 3a illustrates a traditionally conditioned single module transformer model. Our recipe generating system is dependent on two sources: the image features *eI* and ingredients embeddings *eL*. Hence, we want our attention to giving importance to both modalities simultaneously for guiding the instruction generation process. To that end, we have three different fusion strategies (depicted in Figure 3):

– **Concatenated attention**. This strategy first concatenates both ingredients *eL* and image *eI* embeddings. After that, attention is applied over the combined embeddings.

– **Independent attention**. This strategy includes two attention layers to handle the bi-modal conditioning. In this case, one-layer attends over the image embedding *eI* , whereas the other attends over the ingredient embeddings *eL*. The output of both attention layers is combined through summation operation.

– **Sequential attention**. This strategy sequentially attends over the two conditioning modalities. In our system, we study two orderings: (1) first image embeddings *eI* and then over ingredient embeddings *eL* and (2) ingredients first where the order is flipped and we first attend over ingredient embeddings *eL* followed by image embeddings *eI.*

3.2 Optimization

Our recipe system is trained in two stages. The first stage consists of a pre-training image encoder and ingredients decoder. Then, the next stage consists of training ingredient encoder and instruction decoder, along with that it will minimize the negative log-likelihood and adjust the values of

learnable parameters. During the training period, the instruction decoder takes as input the ground truth ingredients. Except for the set transformer, all transformer models are trained with teacher forcing [17].

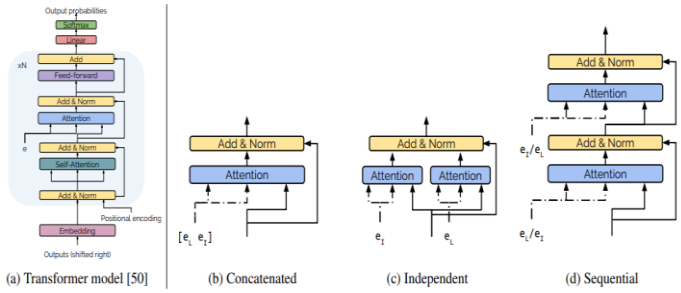

Fig-3: Attention Strategies

4. EXPERIMENT

This section explains the dataset and detailed description of execution which is followed by an exhaustive analysis of strategies used for the cooking instruction transformer. Next, it compares proposed ingredient prediction models to previously introduced baselines. It also shows a comparison between our inverse cooking system and retrieval-based model.

4.1 Dataset

We have used the Recipe 1M dataset to obtain food images and recipes for training our model. The recipes were scraped from many common and popular cooking websites and processed through a pipeline that extracted relevant text from the raw HTML, downloaded linked images, and assembled the data into a compact JSON schema. For extraction procedure, excessive whitespace, HTML entities, and non-ASCII characters were removed from the recipe text. The ingredient strings in this dataset are fundamental contribution work to make it usable for machine learning tasks.

Thanks to the latest advances in the field of science and technology, numerous people have got access to the internet. Many social media sites and simple websites have become platforms where users upload and share images and videos about various topics. Food is one such topic. They have essentially become data containers. Many search engines, such as Google and Bing go through a large number of images, websites, videos, or any other content that matches a text query. In amassing this dataset, the Google search engine was used.

Initially, 50 million images were downloaded, 5 images per recipe. The title of each recipe was used as a query for obtaining the images. Top 50 results from the query result were downloaded and added to the dataset. For this task, publicly available python libraries were used. Next, corrupted and inappropriate images were filtered out.

The dataset includes approximately 0.4% duplicate recipes and, apart from the duplicate recipes, about 20% of recipes have non-unique titles but differ by a median of sixteen ingredients. 0.2% of recipes share equivalent ingredients but are relatively straightforward, having a median of six ingredients. Approximately half of the recipes did not have any associated images in the initial data collection from recipe websites. After the data extension phase, around 2% of the recipes are left without any corresponding images. Exact duplicates were removed after careful observation, using ResNet18 as a feature extractor.

Around 70% of the data is labelled as training, and the rest of the dataset is split equally between the test and validation sets. During the dataset extension, an intersection dataset was created, in order to have a fair comparison of the experimental results on both the initial and the extended versions of the dataset.

The contents of the dataset can be classified into two layers. The first layer contains basic information - a title, a list of ingredients, and a sequence of instructions for preparing a dish, in text format. The second layer adds to the first layer, by including all the corresponding images, in JPEG format.

The average recipe in the dataset comprises nine ingredients that are transformed over the course of ten instructions. It can be observed that the distributions of data are heavy-tailed. For example, of the 16k ingredients identified as unique, only 4,000 account for 95% of occurrences. At the low end of instruction count, one will find the 'Combine all ingredients' instruction. At the other end are lengthy recipes and ingredient lists associated with recipes that include sub-recipes, which further increase the number of instructions.

A similar issue of outliers exists also for images: since several of the included recipe collections are curated from user-submitted images, popular recipes like chocolate cake have many more images than the typical recipe. The number of unique recipes with corresponding food images in the initial process of the data collection phase was 333K. After the data extension phase, this number jumped to 1 million. On average, the Recipe1M+ dataset contains 13 images per recipe whereas Recipe1M has less than one image per recipe. Hence, we are going to use the extended dataset to attain better results.

4.2 Implementation Details

All the neural network models are implemented using the Tensorflow2 framework. We have resized images into 256 pixels and took a random crop of 224x224 for training. For evaluation, we select 224x224 pixels from the center. We used a transformer with 16 blocks and 8 multi-head

attention, each one having dimensionality of 64 for an instruction decoder. For the ingredient decoder, we use a transformer with 4 blocks and 2 multi-head attention, each one with the dimensionality of 256. We have used the last layer of ResNet-150 to obtain image embedding. The dimension of image and ingredients embedding is 512. The maximum limit for ingredients per recipe is set to 20 and truncate instructions to a maximum of 150 words. The models are trained with Adam optimizer [18] until early-stopping criteria are met (using patience of 50 and monitoring validation loss).

4.3 Recipe Generation

We have introduced multi-modal architecture with different types of attention strategy. Table 1 shows the perplexity of each attention strategy model on the validation dataset. From the table, we can infer that the result of independent attention is worse and it is followed by sequential attention. This shows the incompetence of these strategies to predict the expected output. However, the best result is shown by the concatenation attention i.e. 8.50 because it is easily adaptable to give importance to each modality whereas independent attention is forced to include importance from both the model encoder. Hence, we have used the concatenation attention model on our test dataset to produce relevant results.

Model	Perplexity
Independent	8.59
Sequence image first	8.53
Sequence Ingredient first	8.61
Concatenated	**8.50**

Table 1: Recipe perplexity

We have compared our system with the other two models. One of them is an image-to-sequence instruction system that directly generates instruction from an image feature (I2R) while the other model removes the visual features and predicts the sequence instruction from ingredients (E2R). By improving both I2R and E2R baseline, our system has achieved a perplexity of 8.51 on the test dataset. It has conquered the other two models having perplexity of 8.67 (L2R) and 9.66 (I2R) from which we can infer the importance of ingredients in predicting instructions. Finally, we have trained our model and analyzed the results. We notice that generated instructions have an average of 9.21 sentences containing 9 words each, whereas original instructions have an average of 9.08 sentences of length 12.79.

4.4 Ingredient Prediction

Under this section, we compare our ingredient prediction approaches to previously described models, with the goal of whether ingredients should be considered as lists or sets. We have used models from the multilabel classification literature as baselines, and then fine-tune them for our purposes. On one side, we have models set up on feed-forward convolutional networks, which are trained to predict sets of ingredients. We have conducted experiments by considering several losses, namely binary cross-entropy, soft intersection over union as well as target distribution

Model	IoU	F1
ResNet18	17.85	30.30
Inception V2	26.25	41.58
Resnet50	27.22	42.80
Resnet150	**28.84**	**44.11**
TF*list*	29.48	45.55
TF*list* + shuffle	27.86	43.58
TF*set*	**31.80**	**48.26**

Table 2: IoU and F1 of Ingredient models

cross-entropy. On the other side, we have sequential models that predict the ingredients as lists by improving order and using dependencies among elements. Finally, we have used recently proposed models in which set prediction is coupled with cardinality prediction to determine which elements to include in the set [15] because we see that models that exploit dependencies, consistently increase ingredient's F1 scores, and strengthen the significance of modeling ingredient co-occurrences.

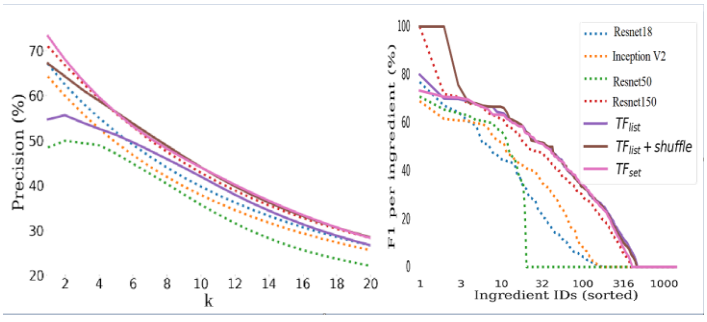

Fig-4: Ingredient prediction results

5. EVALUATION

Under this section, we have compared our proposed recipe generation system with the retrieval system, which we use to search recipes in the entire test set for a fair comparison.

Ingredient prediction evaluation. The recipe retrieval model in [3] is used as a baseline and compare it with our best ingredient predictions models, namely ResNet-150 and TF*set*. The retrieval model is referred to as R*i2lr*, which is trained by joint embeddings of images and recipes (title, ingredients, and instructions). Therefore, for the task of predicting ingredients in a given recipe, we have used the image embeddings to retrieve the closest recipe and report metrics of the ingredients for the retrieved recipe. We also consider another alternative retrieval system which is trained by joint embeddings of images and ingredients list (without considering title and instruction and refer to it as R*i2l.* Table3 provides the obtained results on the Recipe1M test dataset. The R*i2lr* model performs much better than the R*i2l* model, which shows the importance of complementary information present in the ingredient's dataset. However, our proposed model has outperformed both the retrieval-based model by a large margin (e.g. TFset outperforms the R*i2lr* retrieval baseline by 12.26 IoU points and 15.48 F1 score points), which signifies the superiority of our models.

Model	IoU	F1
R*i2l* [3]	18.92	31.83
R*i2lr* [3]	19.85	33.13
ResNet150 (ours)	29.82	45.94
TF*set* (ours)	32.11	48.61

Table 3: Comparision of IoU and F1 scores

Model	Recall	Precision	Accuracy
R*il2r*	31.92	28.94	30.35%
Inverse Cooking [25]	75.47	77.13	76.30%
Ours	**77.89**	**79.08**	**78.48%**

Table 4: Precision and Recall of ingredients Model

Recipe generation evaluation. We have compared our proposed system of instruction decoder which generates instructions given an image and ingredients with a retrieval model. For an unbiased comparison, we have retrained the retrieval model to find the cooking instructions by providing both ingredients and images. In our evaluation, we have considered the ground truth ingredients as a reference and calculated recall and precision w.r.t. the ingredients that appear in the obtained instructions. The recall gives the percentage of ingredients in the reference that is present in the output instructions, whereas precision provides the percentage of ingredients appearing in the instructions that also appear in the reference. Table4 displays a comparison between our model and the retrieval system. According to the results, ingredients appearing in our proposed generated instructions have outperformed the recall and precision scores of the ingredients in retrieved instructions.

6. CONCLUSION

In this research paper, we have implemented an image-to-recipe generation system, in which a particular food image, when fed into the system, will yield four most probable recipes consisting of its Title, the constituent ingredients, and sequence of cooking instructions. First, it will determine the possible sets of ingredients used, from the food images, showing the modeling dependencies, and, along with that, we have explored instruction generation conditioned on the images and inferred ingredients, which highlights the importance of implementing both modalities at the same time. Further, we hope that this paper will assist in the formation of more automated tools for food and recipe understanding. Finally, user study results confirm the complication of the task and demonstrate the supremacy of our model against state-of-the-art image-to-recipe retrieval systems.

Acknowledgements:

We would like to express special thanks to CSAIL mit.edu for making Recipe1M publicly available dataset. We would also want to express our gratitude to my friends, which also helped me in doing a lot of research and for their fruitful comments and suggestions.

7. REFERENCES

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017.

[3] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. Learning cross-modal embeddings for cooking recipes and food images. Training, 720:619-508, 2017.

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

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

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

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

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

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

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015

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

[12] Claude Fischler. Food, self and identity. Information (International Social Science Council), 1988.

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

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

[15] Claude Fischler. Food, self, and identity. *Information (International Social Science Council)*, 1988.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[17] S Hamid Rezatofighi, Anton Milan, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint learning of set cardinality and state distribution. AAAI, 2018

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[19] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. Neural Comput., 1(2), June 1989.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[21] Weiqing Min, Bing-Kun Bao, Shuhuan Mei, Yaohui Zhu, Yong Rui, and Shuqiang Jiang. You are what you eat: Exploring rich recipe information for cross-region food analysis. IEEE Transactions on Multimedia, 2018.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015.

[23] Sara McGuire. Food Photo Frenzy: Inside the Instagram Craze and Travel Trend. https://www.business. com/articles/food-photo-frenzy-insidethe-instagram-craze-and-travel-trend/, 2017. [Online; accessed Nov-2018].

[24] Simon Jegou, Michal Drozdzal, David Vazquez, Adriana´ Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In CVPR-W, 2017.

[25] Amaia Salvador, Michal Drozdzal, Xavier Giro-i-Nieto, Adriana Romero. Inverse Cooking: Recipe Generation from Food Images. CVPR, 2019