Project Report : CS 7643

Stacy Liu, Juan Reyes, Steve Zheng
Georgia Institute of Technology
sliu836@gatech.edu, jreyes70@gatech.edu, szheng319@gatech.edu

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous CVPR abstracts to get a feel for style and length. The abstract section should contain a brief summary of your work that includes the problem statement, proposed solution and results.

1. Introduction/Background/Motivation

With the increasing popularity of visual-centric social networks like TikTok and Instagram, there is a growing desire to identify enticing food dishes and recreate them in our own kitchen. However, merely relying on the image of the dish does not provide insights into how to prepare it or ingredients. This project aims to tackle the challenge of generating recipes from just the dish image.

Since early 2023, LLMs like ChatGPT are undoubtedly the current state tool to answer such prompts. We tested uploading the following image from a roast chicken recipe on Epicurious [3] to GPT 4 and asked it to generate a recipe:

Figure 1: Miso-Butter Roast Chicken With Acorn Squash Panzanella

GPT 4 correctly generated a roasted chicken recipe, and also accurately identified other ingredients like squash (although not "acorn squash" specifically), red onion, and gravy. It did miss the bread chunks and apple slices, which exist in the actual recipe and are identifiable in the image by a human (with some effort). Overall, GPT 4 is likely sufficient for most use cases of image to recipe generation, though it may miss some of the restaurant's "secret sauce".

In order to solve the dish identification and recipe generation problem, the model developed by Facebook Research [4] was considered as the baseline model for our exploration. This model was the state of the art prior to the introduction of multimodal LLM models such as ChatGPT4. This model describes an approach using an image encoding CNN into a transformer. The code leverages a pre-trained CNN image model, such as the Resnet models, to first extract features from the input image. Then, it uses a decoder to generate an ingredients list, and then a final encoder-decoder attending to the image features and predicted ingredients to output a recipe title and instructions. The below diagram from the paper summarizes the architecture:

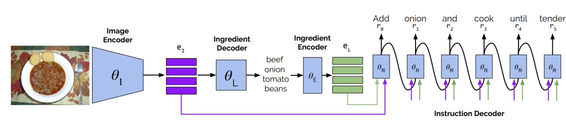

Figure 2: Facebook Research Model Architecture

The state-of-the-art dataset for food images and recipes had been Recipe1M, which was the underlying data for the Facebook Research model. Unfortunately, this dataset does not appear to be publicly available, and we received no response when reaching out to the dataset owner.

Instead, we chose this Kaggle dataset [3] for our project, because it also contains a collection of recipes and accompanying images. The dataset consists of a CSV file with 13,501 rows of food recipes and a zip file containing images corresponding to the recipes in the CSV; there is one image per recipe. Each row in the CSV includes the recipe title, a list of ingredients, instructions on making the recipe, and the name of the .jpg file in the zip corresponding to the row. The dataset was created by scraping the Epicurious website; no further details on the scraping or curation process was provided by the authors.

In order to leverage the Facebook Research model, we had to manipulate the Kaggle data to fit the Recipe1M schema. The biggest difference was that the Recipe1M data seemed to have a processed ingredients list that only contained the ingredients without modifiers (e.g. "X tsp of...", "freshly grounded...", "pinch of..."). This ingredients list facilitated the building of an ingredients vocabulary that flowed into the model. The ingredients in the Kaggle dataset contained modifiers. To extract just the ingredients, we made an assumption that the recipe instructions would mention the ingredients in unmodified form. For example, we assumed the ingredients list would contain "3-4 lb. whole chicken", but the recipe would refer to the ingredient as just "chicken". Based on this assumption, we removed stopwords and extracted words that appeared in both the ingredients and the recipe instructions to create the list of unmodified ingredients. This recipe held true for most of the data, but ultimately we were left with 10k rows that seemed suitable.

2. Approach

2.1. Baseline model

Considering computational and temporal restrictions on the project, the model pretrained by Facebook Research [4] was considered as a baseline to evaluate the performance on the Kaggle dataset [3], providing a starting point for the performance analysis. The primary intention was to explore the ability of the base model, a model previously trained by Facebook on the Recipe1M dataset (which has more than 1 million data images) to make correct predictions. This baseline not only represents the state-of-the-art prior to the adoption of large language models (LLMs), but also offers a unique opportunity to evaluate such a top performing model when applied to a smaller dataset.

For greater detail on the characteristics of the model training, the supplementary material of the paper [4] can be consulted. Nevertheless, a small description of the model is presented below. The base model architecture extracts image representations in the encoder with a ResNet-50 convolutional network [1]. For the decoder of the instructions, a transformer with 16 blocks and 8 multi-attention heads is used. For the ingredient decoder, a transformer is used with 4 blocks and two multi-attention heads, each with a dimensionality of 256. The model considers a dimension of the embeddings of 512. The base model has a limit of 20 ingredients per recipe and a maximum of 150 words per recipe. The model considers the Adam optimizer with stopping criterion for its training.

For our project, we trained four different models, all based on the architecture of the Facebook Research base model, but with specific variations in the encoder structure. The goal of this variation was to discern how differences in encoder architecture affect model performance. For this, four convolutional model architectures were selected, all derived from the ResNet model, known for its robustness in visual classification tasks. The models implemented were ResNet50, ResNet18, ResNet101 and ResNet152. Each of these pre-trained convolutional models was trained with the same data set and hyperparameters such as batch size, learning rate. Given the limitations in computational resources, the models were trained for 50 epochs, as training for longer periods was not possible due to connection problems on Google Colab Pro on a T4 GPU server. This experiment compares the effect of the architecture and complexity of the encoder model on the recipe prediction task. This variation in depth can impact the ability to identify the elements of the plate and therefore the performance of the model. Additionally, by training the models on the dataset and specific vocabulary it is possible to expect a performance close to, but not exceeding, the base model due to limitations in the amount of training data.

2.2. Image to Caption model

Given the results of the baseline model (see Section 3), we decided to experiment with a simpler model. Rather than trying to generate the title, the ingredients, and the recipe instructions from an image, we simplified the generation to just the caption. The idea being that if we can train a model to generate the dish name, that dish name can be passed onto a LLM to generate the recipe.

Following the Image Captioning tutorial by Magnus Pedersen [2], we used the VGG16 model that has been pre-trained for classifying images. Instead of using the last classification layer, the output of the previous layer is redirected to a RNN decoder. The RNN decoder is comprised of 3 layers of Gated Recurrent Units (GRU), similar in nature to LSTM which has a gating mechanism to input or forget certain features. Figure 3 shows the flowchart of the Image Captioning architecture:

For the RNN decoder, we trained using the same Kaggle dataset. The training image data was processed through the pre-trained VGG16 model and the transfer-values were saved in a cache file to speed up training. The training caption data was processed in two steps: 1) convert text to integer tokens, 2) convert integer tokens into an embedding layer. Finally, the model was put together using Keras layers consisting of an embedding layer, 3 GRU layers, and a final dense layer. The loss function we used was a sparse cross-entropy loss since the data set is comprised of integer-tokens that maps to 10,000 elements. Using sparse cross-entropy loss eliminated the need to convert the dataset to a sparse one-hot encoded arrays since that's done internally within the loss function.

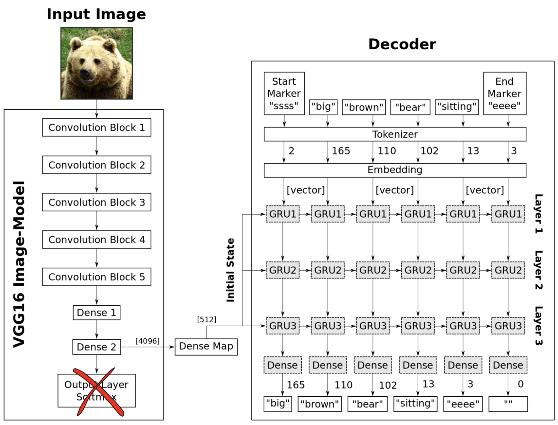

Figure 3: Architecture of Image-to-Caption model, comprising of a pre-trained VGG16 encoder and a RNN decoder

In order to leverage this Image-to-Caption model, we had to manipulate the Kaggle data to fit this architecture. First the images had to be reshaped to be of size (224, 224, 3) in order to be processed by the VGG16 model to generate transfer-values for later steps. Second, the images in the original training set had at least 5 captions per image. In our case, the Kaggle data only has one caption per image so modification was done so that the batches would be generated correctly for model training.

3. Experiments and Results

3.1. Baseline model

In order to quantify the performance of the experiments with the base model and the trained models both quantitatively and qualitatively, predictions of the titles of the dishes and ingredients were made on a set of 2,100 testing images. For the quantitative evaluation of the models, a similarity score was computed for both the name of the dish and the list of ingredients. Additionally, the Jaccard metric was used to estimate the similarity between the actual ingredient lists and the predicted ingredient list. Qualitatively, we have outputted a word cloud to visualize the most frequent ingredients predicted by the models.

To estimate the similarity score, a SentenceTransformer model (all-MiniLM-L6-v2) was used. This model converts each predicted dish name into a high-dimensional vector. Once the phrase has been transformed, the cosine similarity is calculated. Values close to 1 indicate high similarity while values close to 0 indicate low similarity. The second similarity score used was the Jaccard similarity. This score is computed as the size of the intersection divided by the size of the union of two sets. See this page for more details on the two metrics. A Jaccard index close to 1 indicates a similarity between the list of ingredients, while a value of 0 indicates that real ingredients in the predicted ingredients list are not found.

In Table 1 we present the average plus/minus one standard deviation for a number of metrics using the set of 2,100 test images, namely: the similarity score of the dish names, the ingredients list, the number of predicted ingredients, and difference in number of ingredients. Note that the Image to Caption model only outputs the dish name, so the ingredients list metrics were not applicable.

As can be seen in Table 1, the base model (Meta ResNet-50) presents the highest dish name similarity score, indicating that, of the base models, it is the model with the greatest capacity to capture semantic similarity in the names of dishes. This was to be expected since the model was trained on a dataset with a size at least 10 times larger than the Kaggle dataset. The ResNet-152 model outputted the highest similarity score regarding the correct identification of the ingredients of the dishes.

On the other hand, assessment using Jaccard similarity offers a different perspective. Here, the ResNet-50 model shows the highest value close to 0.2001, suggesting a better ability to identify common ingredients between the predicted and actual lists. It is striking that the base model has the lowest Jaccard similarity value. This may be due to a difference between the ingredients dictionary in the Recipe1M dataset compared to the dictionary in the Kaggle dataset.

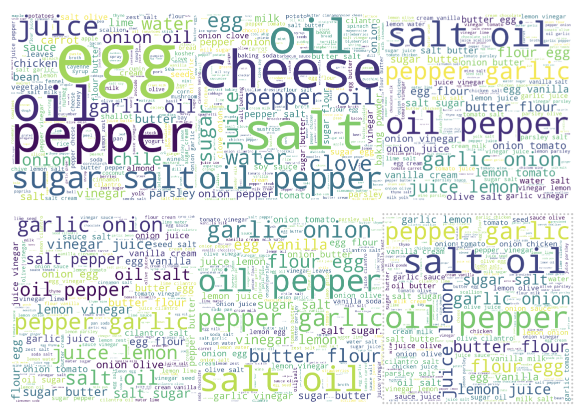

Figure 4: Ingredients word cloud for (from top-bottom, left-right): true ingredients, predicted ingredients for Meta model, predicted ingredients for ResNet-18, predicted ingredients for ResNet-50, predicted ingredients for ResNet-101, and predicted ingredients for ResNet-152

As can be seen in the word clouds in Figure 4, there are predominant or common ingredients throughout the dishes such as ”salt”, ”pepper”, ”oil”, ”garlic”, ”sugar”, and ”butter” which is to be expected since they are common elements across different food cultures. Another element to highlight is the variety of ingredients reflected in the word

Model	Model Params	Dish Name Sim	Ingredients Sim	Ingredients Jaccard Sim	Predicted Ingrs	Diff in Ingrs
Meta (ResNet-50)	103,779,021	0.398 ± 0.179	0.535 ± 0.142	0.166 ± 0.116	6.155 ± 2.097	6.145 ± 5.855
ResNet-18	68,238,158	0.312 ± 0.132	0.542 ± 0.136	0.196 ± 0.112	8.900 ± 1.174	3.400 ± 5.722
ResNet-50	81,356,110	0.325 ± 0.135	0.544 ± 0.133	0.200 ± 0.109	9.001 ± 1.571	3.299 ± 5.752
ResNet-101	100,348,238	0.317 ± 0.135	0.536 ± 0.133	0.196 ± 0.111	8.318 ± 1.517	3.982 ± 5.817
ResNet-152	115,991,886	0.320 ± 0.137	0.558 ± 0.137	0.197 ± 0.111	9.715 ± 2.172	2.585 ± 5.734
Image-to-Caption (20 Epochs, 3 GRU Layers)	1,264,564	0.289 ±0.124	N/A	N/A	N/A	N/A
Image-to-Caption (20 Ep, 6 GRU Layers)	17,373,456	0.290 ±0.122	N/A	N/A	N/A	N/A
Image-to-Caption (50 Ep, 6 GRU Layers)	17,373,456	0.288 ±0.124	N/A	N/A	N/A	N/A

Table 1: Model number of parameters, average similarity score for dish name and ingredient list, average Jaccard similarity for ingredient list, number of predicted ingredients and difference of predicted vs true number of ingredients

clouds, which suggests that the models have been able to learn a wide range of elements present in the recipes.

In Table 2, we present the predicted dish names for 6 images of a demo dataset that was unused in the model training or validation. As expected from the base Facebook model, this model manages to make coherent predictions of the images and predicts the type of dish such as pasta-based dishes and enchiladas. In contrast, the models trained on the Kaggle dataset tend to repeat ingredients in the prediction of titles such as “lemon”, “onion” and/or “chicken”. Although the models have learned to identify some ingredients, the models have a limited ability to differentiate dishes beyond the most common ingredients.

In Table 3 you can see the list of ingredients predicted by the different models for the same images used to predict the dish name. The base Facebook model, as with the names of the dishes, manages to identify central ingredients of the dishes such as “Shrimp”, “Avocado” and “Eggplant”. However, it also predicts ingredients that may not be visible on dishes such as “mayonnaise” or “beans” that the model could have included based on training patterns and their frequency with other ingredients.

On the other hand, ResNet models seem to be predicting common elements ”salt”, ”oil”, ”pepper”, ”garlic”, ”onion” which, as previously mentioned, indicates that the models have managed to learn common patterns of the different dishes during training. Nevertheless, unlike the Facebook model, these models have not been able to capture the central ingredients of the dishes in question by predicting generic and common ingredients such as those mentioned previously. Although the ResNet models predict a greater number of ingredients as presented in Table 3 and the Jaccard similarity score is similar to the Facebook model, it can be concluded that their performance in predicting the central ingredient of the dishes is limited relative to base Facebook model. Based on these results, it can be concluded that there is ample room for improvement in the performance of these models that have been hampered by limitations of computational resources and access to datasets with a greater amount of training data.

3.2. Image to Caption Model

The following experiments were performed on the base Image-to-Caption model [2]:

1. Increase to 6 layers of GRU
2. Increase to 6 layers of GRU and 50 epochs

In the first experiment, we doubled the number of GRUs in the RNN decoder. The idea is that with more GRUs, the RNN should be able to learn more complex patterns in the data. Since GRUs can capture and remember information from previous time steps, having more units would increase the network’s memory and processing capabilities. Based on the similarity results, this is indeed the case. The title similarity score increased from 0.2892 to 0.2901 as seen in Table 1.

In the second experiment, we increased the epochs from 20 to 50. The idea is that with more epochs, the RNN will have additional opportunities to learn from the data. However, due to the limited size of the Kaggle dataset, this resulted in overfitting. The model learned the noise and idiosyncrasies in the training data rather than generalizable patterns, which led to poor performance on test data.

While the Image-to-Caption model (0.290 title similarity score) did not perform as well as the Facebook model (0.398), it was also a much lighter model as shown in Table 1. With less than 20% of the Facebook model’s parameter, it was able to get semi-close to the title generated by that model. Some of the closest generated captions are shown in Figure 5:

Once the captions are generated, they are then fed into GPT4 to generate the ingredients and recipe. We note that the captions generated by Image-to-Caption are sometimes “creative”. For example, “cherry beer burger” was generated for what is clearly avocado toast. Despite this, the combination of Image-to-Captions and GPT4 resulted in much better ingredient and recipes than just the ResNet models. For example, for Image 1, Image-to-Caption generated “spaghetti with mussels on and grapes” and GPT generated “Spaghetti, Fresh mussels, Red or green grapes, Garlic, Olive oil, White wine, Parsley, Salt, Pepper” as the ingredient. This is clearly much more detailed than the ones

Model	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
Dish photo	1.jpg	2.jpg	3.jpg	4.jpg	5.jpg	6.jpg
Meta (ResNet-50)	Garlic shrimp scampi	Avocado egg salad sandwich	Grilled eggplant and zucchini	Penne with caramelized onions	Easy pancakes	Chicken enchiladas
RResNet-18	Grilled onions with lemon and lemon	Grilled red pepper and lemon	Grilled onions with lemon and lemon	Grilled chicken with lemon and lemon	Vanilla ice cream	Chicken with chicken and chicken
ResNet-50	Grilled red onion salad	Spiced fried eggs	Roasted red onion and garlic	Grilled red onion and onion	Chocolate chip cookies	Tomato and tomato salad
ResNet-101	Grilled onion and red onion	Grilled red onion	Fried eggs	Roasted garlic and parmesan	Chocolate chocolate cake	Grilled fish with olive oil
ResNet-152	Grilled chicken with fried eggs	Eggs with lemon and lemon	Grilled onion and onion	Grilled onion and onion	Eggs with buttermilk and eggs	Chicken with chicken and chicken
Image-to-Caption (20 Epochs, 3 GRU Layers)	spaghetti with mussels on and grapes	cherry beer burger	grilled leg cake with bakewell cheese	pulled brisket with hot paste and tea chiles	peach cheesecake with orange syrup	pumpkin pie with sour mincemeat crust
Image-to-Caption (20 Epochs, 6 GRU Layers)	almonds with mussels meatballs and grapes	cherry beer burger	grilled leg cake with sugarhoney cheese	pulled walnuts with hot paste and wings watermelon	cakes cheesecake with orange syrup	oldfashioned pie with sour real crust
Image-to-Caption (50 Epochs, 6 GRU Layers)	grilled pork shoulder with marinated lemon	grilled cabbage croquettes	dryrubbed turkey breast	penne with ramp pesto	cherry french chicken with lemon tomatoes	rigatoni with eggplant and pine nut crunch

Table 2: Predicted dish name for images based on the demo dataset

Model	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
Dish photo	1.jpg	2.jpg	3.jpg	4.jpg	5.jpg	6.jpg
Meta (ResNet-50)	shrimp, pepper, butter, clove, oil, salt, pasta, parsley	avocado, bread, tomato, mayonnaise, pepper, egg, salt	zucchini, oil, salt, eggplant, pepper	cheese, pasta, pepper, onion, oil	sugar, flour, egg, salt, butter, baking_powder, milk, oil	cheese, tortilla, onion, cream, enchilada_sauce, chicken, chili, beans, pepper, tomato, cumin
ResNet-18	salt, oil, pepper, garlic, onion, butter, juice, lemon	salt, oil, pepper, garlic, onion, lemon, egg	salt, oil, pepper, garlic, lemon, butter, onion, egg	salt, oil, pepper, garlic, onion, butter, thyme, lemon, wine, parsley	salt, butter, flour, egg, vanilla, cream, milk, sugar	salt, oil, pepper, garlic, onion, butter, tomato, juice, chicken, wine, parsley, lemon
ResNet-50	oil, salt, pepper, garlic, onion, olive, vinegar, juice, lemon	salt, oil, sugar, pepper, garlic, egg, flour	salt, oil, pepper, garlic, onion, olive, lemon	salt, oil, pepper, garlic, onion, sauce	salt, sugar, butter, egg, flour, vanilla, cream	salt, oil, pepper, garlic, onion, egg, vinegar, juice, lemon, tomato
ResNet-101	oil, garlic, onion, water, butter, vinegar	salt, oil, pepper, garlic, vinegar, lemon, onion, olive	salt, oil, butter, onion, egg, flour	salt, oil, pepper, garlic, onion, butter, egg, flour, lemon	sugar, salt, butter, flour, egg, vanilla, chocolate, pan, soda, cream, cinnamon	salt, oil, pepper, garlic, onion, olive
ResNet-152	oil, salt, garlic, pepper, sauce, egg, olive, flour, soy	salt, oil, pepper, vinegar, sauce, egg, lemon, juice, flour, vanilla	oil, juice, lemon, garlic, onion, water	salt, oil, pepper, garlic, butter, lemon, onion, flour, parsley	salt, oil, butter, egg, flour, cream, olive	salt, oil, pepper, garlic, onion, sauce, tomato, olive, egg, chicken
Image-to-Caption (20 Epochs, 3 GRU Layers), GPT4	Spaghetti, Fresh mussels, Red or green grapes, Garlic, Olive oil, White wine, Parsley, Salt, Pepper	Ground beef, Burger buns, Cherry beer, Cherries (fresh or dried), Lettuce, Tomato, Onion, Cheese (optional), Salt, Pepper	Leg meat (chicken or lamb), Cake mix (savory), Bakewell cheese, Eggs, Milk, Fresh herbs (like thyme or rosemary), Salt, Pepper	Beef brisket, Hot paste (chili or harissa), Dried chiles soaked in tea, Onion, Garlic, Tomato paste, Beef stock, Brown sugar, Salt, Pepper	Cream cheese, Graham cracker crust, Fresh peaches, Eggs, Sugar, Vanilla extract, Orange juice (for syrup), Sugar (for syrup), Orange zest	Pumpkin puree, Sour mincemeat, Flour (for crust), Butter (for crust), Eggs, Evaporated milk, Brown sugar, Cinnamon, Nutmeg, Ginger, Cloves
Image-to-Caption (20 Epochs, 6 GRU Layers), GPT4	Almonds, Mussel meat, Breadcrumbs, Egg, Garlic, Red or green grapes, Olive oil, Parsley, Salt, Pepper	Ground beef, Burger buns, Cherry beer, Cherries (fresh or dried), Lettuce, Tomato, Onion, Cheese (optional), Salt, Pepper	Leg meat (chicken or lamb), Cake mix (savory), Sugarhoney cheese (or a similar sweet-flavored cheese), Eggs, Milk, Fresh herbs, Salt, Pepper	Walnuts, Hot paste (chili or harissa), Chicken wings, Watermelon (cubed or balled), Onion, Garlic, Brown sugar, Salt, Pepper	Cream cheese, Cake pieces (any flavor), Graham cracker crust, Fresh oranges, Eggs, Sugar, Vanilla extract, Orange juice (for syrup), Sugar (for syrup), Orange zest	Filling of your choice (fruit or custard), Sour cream (for the crust), Flour (for crust), Butter (for crust), Sugar, Eggs (for filling and crust), Salt, Spices (according to filling)
Image-to-Caption (50 Epochs, 6 GRU Layers), GPT4	Pork shoulder, Lemons, Olive oil, Garlic, Rosemary, Salt, Pepper	Cabbage, Breadcrumbs, Eggs, Flour, Onion, Garlic, Salt, Pepper, Cooking oil (for grilling)	Turkey breast, Brown sugar, Paprika, Garlic powder, Onion powder, Dried thyme, Salt, Pepper	Penne pasta, Ramps (wild leeks), Parmesan cheese, Pine nuts, Garlic, Olive oil, Salt, Pepper	Chicken (thighs or breasts), Cherries, Cherry tomatoes, Lemon juice, Garlic, Herbs de Provence, Olive oil, Salt, Pepper	Rigatoni pasta, Eggplant, Pine nuts, Garlic, Olive oil, Parmesan cheese, Basil, Salt, Pepper

Table 3: Predicted ingredients list for images based on the demo dataset

```
Original Caption: negroni
Generated Caption:  negroni
Cosine Similarity: 1.0

Original Caption: sweet summer corn soup
Generated Caption:  roasted corn soup
Cosine Similarity: 0.8396082520484924

Original Caption: shamrock shake
Generated Caption:  rum matcha shamrock shake
Cosine Similarity: 0.8266215324401855

Original Caption: our favorite chocolate chip cookies
Generated Caption:  chewy chip cookies
Cosine Similarity: 0.7886237502098083

Original Caption: homemade marshmallows
Generated Caption:  without marshmallows
Cosine Similarity: 0.7707875967025757
```

Figure 5: Top 5 generated captions by best Image-to-Caption model

generated by ResNet (ex. "oil, garlic, onion, water, butter, vinegar"); see Table 3.

4. Experience

4.1. Challenges

We faced a significant upfront challenge in procuring an organized dataset with both recipes and images. When proposing the project, we had envisioned using the Recipe1M data that was used in many related existing implementations. However, as mentioned above, we could not access the Recipe1M data and so ultimately found the current Kaggle dataset, which fit our purpose but only had 10K recipes.

Another challenge was the lack of computing resource to train the models. The baseline model from Facebook Research was trained for 400 epochs by default. However, when we were training the model, our compute environment consistently crashed around the 50-epoch mark even when using paid Google Colab's GPU. Hence, we kept our model training to 50 epochs or lower.

4.2. Conclusion

As shown from our results above, we could not improve on the current state-of-the-art image to recipe generation implementations. We attribute at least some of this to our low volume of training data and computational limitations. Despite this, we enhanced our knowledge of neural networks through deep-diving into several existing implementations. Finally, we came out wholly impressed by what GPT 4 can do with a food image!

5. Appendix

5.1. Team Contribution

See Table 4.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[2] Magnus Erik Hvass Pedersen. Tensorflow tutorials. `https://github.com/Hvass-Labs/TensorFlow-Tutorials`, 2020. 2, 4

[3] sakshidgoel@gmail.com/amoghrajesh1999@gmail.com/tanvipk99@gmail.com. Food ingredients and recipes dataset with images. `https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images/data`, 2020. 1, 2

[4] Amaia Salvador, Michal Drozdzal, Xavier Giro-i Nieto, and Adriana Romero. Inverse cooking: Recipe generation from food images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

Student Name	Contributed Aspects
Stacy	Image to caption to GPT experiments, general dataset research approach planning
Juan	Baseline Facebook Research model tuning and experimentation, general dataset research and approach planning
Steve	Dataset preprocessing to baseline Facebook Research model, baseline model tuning, general dataset research and approach planning

Table 4: Contributions of team members