PROJECT REPORT ON

Food Recipe Generator

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE AWARD OF

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

(2018-2022)

Guide:

Mr. S K Saroj.
Department Of CSE

Submitted By:

Project leader – Aman Jaiswal: 2018021019

Project Member 1 – Piyush Shankar Tiwari: 2018021082

Project Member 2 – Anurag Chaudhary: 2018021029

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY (MMMUT)

Gorakhpur, Uttar Pradesh 273010

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY (MMMUT)

GORAKHPUR, UTTAR PRADESH 273010

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report entitled " Food Recipe Generator " submitted by Aman Jaiswal (2018021019) of Semester VII is a bonafide account of the work done by him under our supervision.

Guide	Head of the Department
Mr. S K Saroj	Mr. A.K. Daniel

Acknowledgements

We would like to sincerely thank our guide **Mr. S K Saroj sir**, Asst. Prof, CSE, for his support and valuable guidance. His timely advice, meticulous scrutiny, scholarly and scientific approach has helped us complete this project on time.

Also, we would like to thank our college, MMMUT, for providing the best facility and support.

We thank each and every staff of the Computer Science Department for lending us help and support. We express our heartfelt veneration to all who had been helpful and inspiring throughout this endeavor. Last but not the least, we thank the almighty for blessing us for completing the project.

Abstract

.

A Food recipe generator which generates recipe from food Image and required ingredients by using Machine Learning and Deep Learning model and there will be facility through a user can purchase ingredients for food.

A Food recommender system which recommends food according to the available ingredients. First our model will classify food from image and then it will extract ingredients and generate recipe for the food. It also recommends food which can be made from available ingredients and also there will be a portal from where user can purchase required ingredients.

For the above, we use machine learning and deep learning algorithm. We will use Spring Boot, Rest API and MongoDB to handle data in backend. For frontend, we will use React along with HTML, CSS, JavaScript.

Contents

1. Introduction ....

2. Related Work ....

3. Methodology ....

4. Model/Methods. ....

5. Exploratory Data Analysis ....

6. Evaluation/Results ....

7. Model Architecture and Parameters ……………………………………..

8. Conclusion/Futo ....

9. References ....

Introduction

Food is fundamental to the human experience. It has deep ties to our health, our livelihood, our emotions, and our culture. More and more these days, people want to learn about what they are eating in order to make better nutritional decisions, but many struggle to find a place to start. Our application aims to empower people to better understand and have control over their food intake in order to help achieve their health goals. The goal of our project is to create a system that accepts an image of a meal as input and outputs closely related recipes as well as nutritional information. Given an input image, Recipe Net returns several related recipes to give options for our user to pick from, and each of these recipes comes with nutritional labels such as under 500 calories,” vegetarian,” and protein heavy in order to guide people to make the best nutritional decisions for themselves. Our platform could help people attain a more sophisticated understanding of health and diet, and enable them to express themselves through new recipes and cooking techniques.

Related Work

In developing our project, we found it extremely helpful to investigate projects aimed at accomplishing similar tasks in order to guide us in the right direction and gain a better sense of what came before us. We found that in the past decade, there has been significant progress in the collection of food-recipe datasets. In the early stages, most works followed a classical recognition pipeline, focusing on feature combination and specialized datasets (mostly Asian cuisine). However, in 2014, Bossard et al. took a step forward and introduced the holistic Food-101 visual classification dataset with 101k images divided equally among 101 classes, setting a baseline accuracy of 50.8%. Two years later in 2016, Chen and Ngo presented another large dataset with 65k recipes and 110k images, but it only covered Chinese cuisine. In 2017, Salvador et al. released the Recipe1M+ dataset, which was a new large-scale, structured corpus of over one million cooking recipes and 13 million food images. To date, this is the largest publicly available collection of food and recipe data, and allows for the ability to train high-capacity models on aligned, multi-modal data. Therefore, this is the dataset we chose to use for our project.

There have also been a variety of approaches used to tackle the food recognition issue. In 2010, Yang et al. proposed to learn spatial relationships between ingredients using pairwise features. However, this was bound only to work for standardized meals. Along with the Food-101 dataset, Bossard et al. introduced a novel method to mine discriminative parts using Random Forests. Random Forests was used to discriminatively cluster super pixels into groups (leaves) and then used the leaf statistics to create features. The researchers applied a distinctiveness measure to the leaves to merge similar leaves, then used a support vector machine (SVM) to classify the food images into categories. A few years later, Herranz et al. proposed an extended multi-modal framework that explores how visual content, context, and external knowledge can be integrated into food-oriented applications. Around the same time, Min et al. presented a multi-attribute theme modeling approach that incorporates attributes such as cuisine style, course type, flavors or ingredient types. The model learns a common space between different food attributes and their corresponding food images. Finally, there have some interesting papers published specifically in the realm of extracting recipes from images. Chen et al. found that the manner of cutting and cooking ingredients play substantial roles in the food’s appearance, and therefore attempted to predict ingredient, cutting, and cooking attributes given a food image. On the other hand, Chang et al. looked at the possibility of several different preparations for one dish by clustering recipes based on their distance.

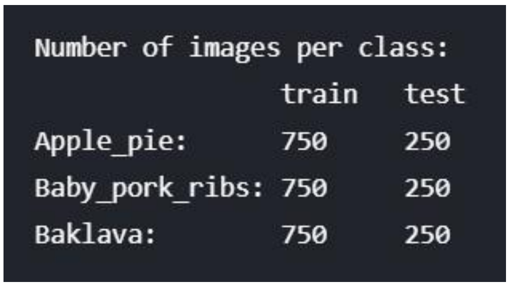
Methodology

Model/Methods

Food 101 is a labelled data set with 101 different food classes. Each food class contains 1000 images. Using the data provided, a deep learning model built on Keras/TensorFlow is trained to classify classes in Food 101 dataset.

Classes: (Apple pie, Baby back ribs, Baklava, …)
Epochs: 100
Batch size: 64

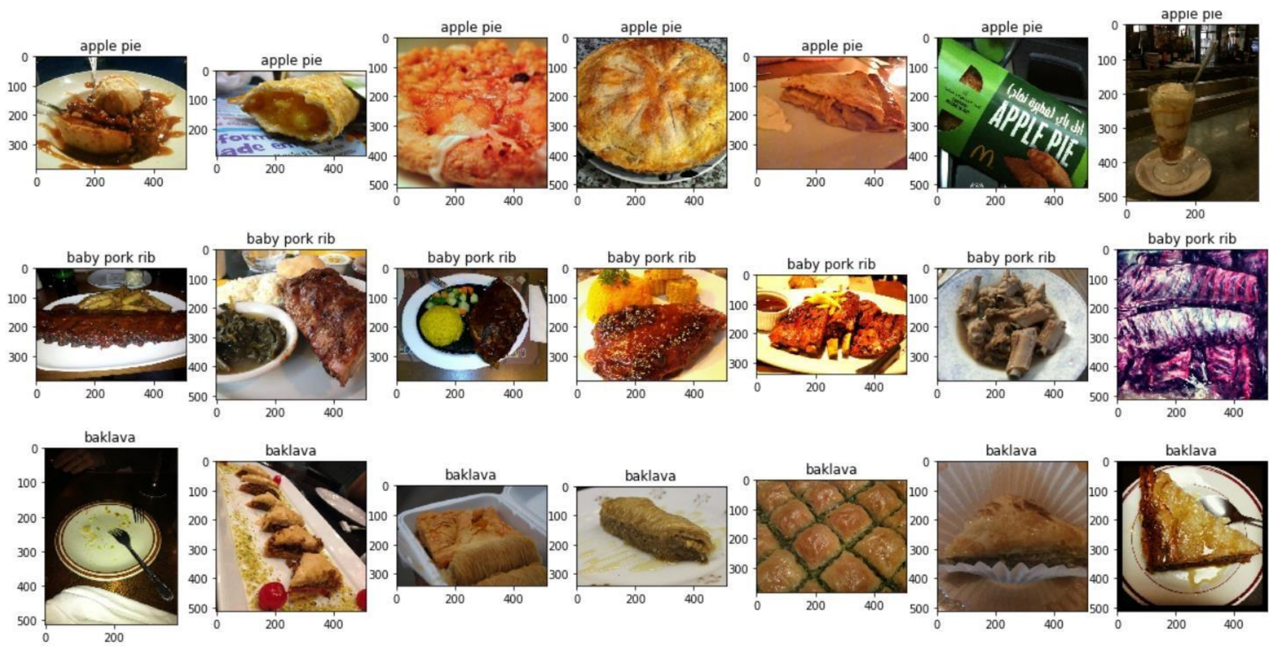
Images are split to train and test set with 750 and 250 images per class respectively.



```
Number of images per class:
                train   test
Apple_pie:      750     250
Baby_pork_ribs: 750     250
Baklava:        750     250
```

Exploratory Data Analysis

Let's preview some of the images.

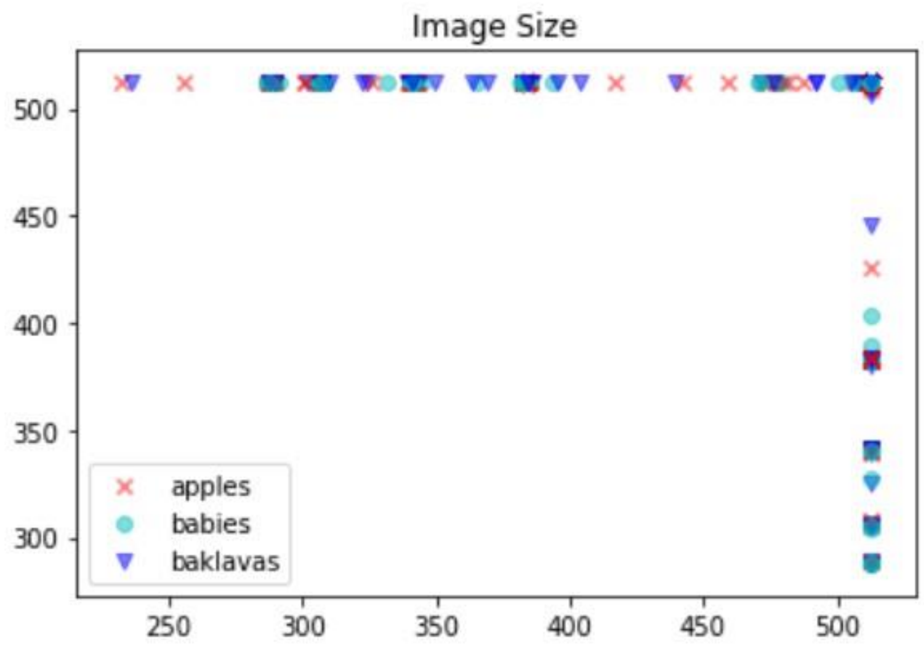


As shown here, the quality of the images is not very good:

- different background
- dissimilar lightings

- wrong labels ○ empty plate in the first pic of the baklava ○ looks more like a pizza than an apple pie for the first apple pie pic ○ looks like an ice cream than an apple pie in the last apple pie pic ○ strange colours of the last baby pork ribs pic

The size of the images is also inconsistent as shown in the height against width plot shown below, but all the images have at least one side with 512 pixels, so we dont have to worry about extremely small images that is pixelated.



To create a convolution neural network to classified the images, Keras Sequential model is used.

Batch normalisation: Tested with batch normalisation layers and removed all dropout layers. It results in faster training and higher learning rates, but it caused more overfitting (large difference between train and test accuracy) than dropout, thus batch normalisation has not been used in this case.

Optimizers: *Adam* final accuracy slightly out-performs *RMSProp* and also converge to minima faster as it's similar to *RMSProp* + *Momentum*.

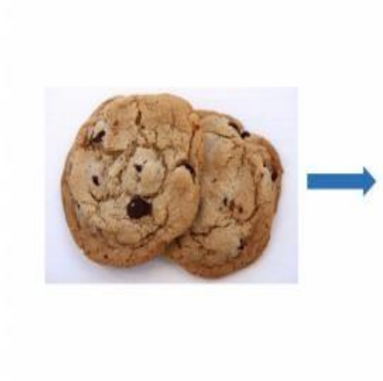
Activation Function: *ReLu* activation used at convolution layers to produce a sparse matrix, which requires less computational power then sigmoid or tanh which produce dense matrix. Also, it reduced the likelihood of vanishing gradients. When a>0, the gradient has a constant value, so it results in faster learning than sigmoid as gradients becomes increasingly small as the absolute value of x increases. *Softmax* activation used at the last layer to assign the probability of each class.

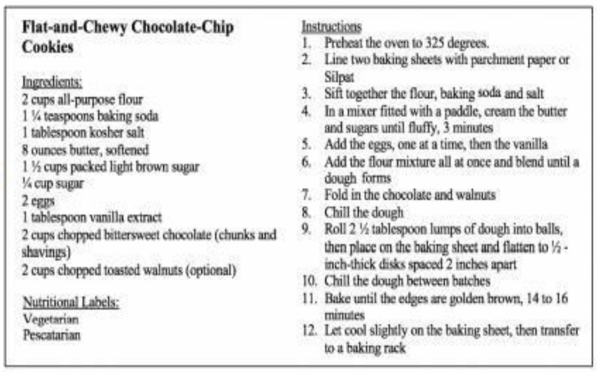
Initializers: Kernal weights are initialized using *He normal* initializers which helps to attain a global minimum of the cost function faster and more efficiently. The weights differ in range depending on the size of the previous layer of neurons and this is a good initializer to be used with *ReLu* activation function.

Regularization: *L2 regularization* is implemented aim to decrease the complexity of the model and minimise overfitting by penalising weights with large magnitudes.

Evaluation/Results

Below are some of the top recipes outputted in running our encodings through ResNet-50 and DenseNet-121 respectively for the input images.





Flat-and-Chewy Chocolate-Chip Cookies

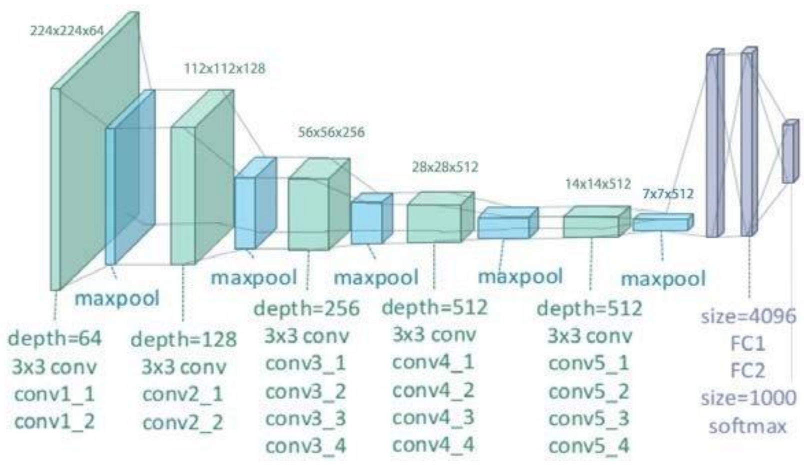
Ingredients:
2 cups all-purpose flour
1 ¼ teaspoons baking soda
1 tablespoon kosher salt
8 ounces butter, softened
1 ½ cups packed light brown sugar
¼ cup sugar
2 eggs
1 tablespoon vanilla extract
2 cups chopped bittersweet chocolate (chunks and shavings)
2 cups chopped toasted walnuts (optional)

Nutritional Labels:
Vegetarian
Pescatarian

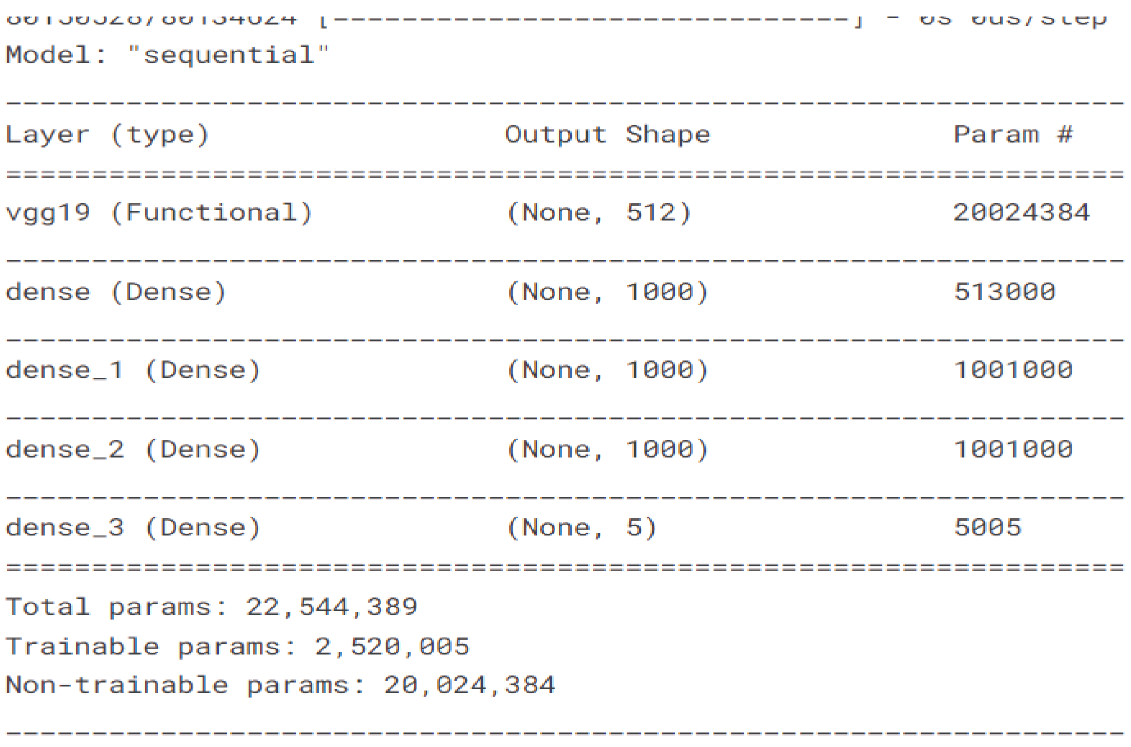
Instructions
1. Preheat the oven to 325 degrees.
2. Line two baking sheets with parchment paper or Silpat
3. Sift together the flour, baking soda and salt
4. In a mixer fitted with a paddle, cream the butter and sugars until fluffy, 3 minutes
5. Add the eggs, one at a time, then the vanilla
6. Add the flour mixture all at once and blend until a dough forms
7. Fold in the chocolate and walnuts
8. Chill the dough
9. Roll 2 ½ tablespoon lumps of dough into balls, then place on the baking sheet and flatten to ½ -inch-thick disks spaced 2 inches apart
10. Chill the dough between batches
11. Bake until the edges are golden brown, 14 to 16 minutes
12. Let cool slightly on the baking sheet, then transfer to a baking rack

To evaluate our model, we tested the relative accuracy/similarity of the encoding of images to the other images attached to the same recipe. Average cosine similarity and average euclidean distance, where values of 1 and 0 respectively are perfect matches of two images, were two metrics we used to quantify this. By using these metrics to measure the similarity of an image to others associated with the same recipe, we can therefore quantify how accurate our found encodings are for an image and thus measure the accuracy of the resulting recipes our model would output. The values found above indicate that using more advanced versions of classification architectures improves our model's ability by providing it with more accurate encodings. Specifically, DenseNet 121 performed significantly better than ResNet-50 and ResNet-101. Overall, these values give confidence in that the recipes recommended are relevant to a given input image. In addition, when setting k = 3, we found that in most cases the first 2 recipes were relatively accurate, but the third recipe was often a little off base. As such, there is room for improvement for higher values of k.

VGG 19 Architecture



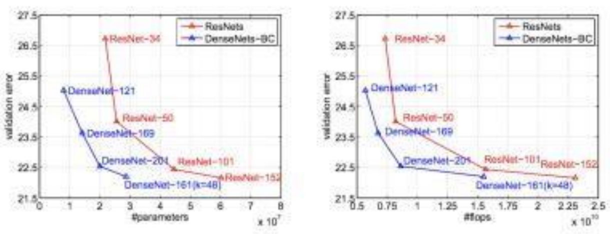
Model Parameter



```
80150528/80134624 [==============================] - 0s 0us/step
Model: "sequential"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Functional)           (None, 512)               20024384

_________________________________________________________________
dense (Dense)                (None, 1000)              513000

_________________________________________________________________
dense_1 (Dense)              (None, 1000)              1001000

_________________________________________________________________
dense_2 (Dense)              (None, 1000)              1001000

_________________________________________________________________
dense_3 (Dense)              (None, 5)                 5005
=================================================================
Total params: 22,544,389
Trainable params: 2,520,005
Non-trainable params: 20,024,384

_________________________________________________________________
```

Conclusion/Future Work

The limitations of time and computing power are ultimately the main constraints in our project. Without such restrictions, future steps could be taken towards implementing models and algorithms that take advantage of these absent limitations in return for a higher accuracy in image classification and closer recipe recommendations. Our model relied on utilizing pre-trained image classification models such as ResNet-50 and DenseNet 121, to handle the initial classification of images. Illustrated below are graphs that plot the relative validation errors for subsequent versions of the DenseNet and ResNet models. Integrating the models with the lowest validation errors would be the obvious next step to guaranteeing improvements to our existing model. Implementing these versions would expand the range of images our model can accurately classify.



In addition, we could train on the entire Recipe1M+ dataset instead of the Recipe 1M subset, which would give us a larger variety of recipes to choose from and improve our performance for higher values of k. Finally, we could expand our project such the user could opt to input a recipe and get a corresponding image. This would likely require the specific instructions and ingredients within the recipes to be encoded, and work alongside our image encodings

- This project contains various technology stack that will bind the full stack and Machine Learning/Deep Learning together.
- We are currently working on the Deep Learning model which will classify the food image. Food 101 dataset has images of food which we already preprocess that into 224 width by 224 height image with 3 channels. That image will fed into the model.
- The project can be improved with various automation by using GAN based model for recipe generation and a dedicated model to detect ingredients of food. But those models are quite complex and research is going on to come up with a feasible solution which will be computationally effective.

- We had collected the data regarding food image to train our model.
- We made a classifier model which will classify image of food.
- We collect the database of food and put it into the mongo dB database.
- We can access the data from mongo dB database to show the list of ingredient of the food inputted by user.
- We can access the data from mongo dB database to show the steps of reciepe of the food.

- Food classification component is complete.

- We have integrated food component in our project.
- We have added more food items to our databases.
- Food recommender system is on the way to complete.
- We are working on the food bucket component.

Reference

Food-101 Dataset

- https://www.kaggle.com/kmader/food41
- https://www.kaggle.com/shuyangli94/food-com-recipes-and-userinteractions?select=RAW_recipes.csv

Weight Initializers:

- Hyper-parameters in Action! Part II — Weight Initializers
- Priming neural networks with an appropriate initializer.

Optimizers:

- An overview of gradient descent optimization algorithms

Batch Normalization:

- Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift
- Glossary of Deep Learning: Batch Normalisation

Activation Functions:

- Understanding Activation Functions in Neural Networks Grad CAM:

- Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradientbased Localization
- Deep Learning with Python Book by Francois Chollet

Regularization:

- L1 and L2 Regularization Methods