DeepFoodie
(Clustering images on basis of Ingredient Embeddings)

Ashutosh Tiwari, Khushboo Singh
{ashutiwa, khusingh}@iu.edu

December 13, 2021

Abstract

1 Introduction

Image classification and Image segmentation has been widely used in recent past. Trivially extending it to identify similar dishes poses an additional challenge due to presence of ingredient and their quantity. Many times we are interested in figuring out similar dishes on basis of ingredients. This has enormous potential, starting from dietary tracking to recommendation engines to allergy prevention. This work presents a novel approach to do same using by learning high dimensional image embeddings and along side integrating the ingredient associated with the dish. The high dimensional embeddings are obtained by image recognition Resnet based model trained on around 13K images whereas the ingredients are encoded as vectors to the ingredients' representation. The rest of the report is organized as follows - Section 2 describes the related work, followed by the Objective in section 3 and Resources in section 4. Final section introduces the methodology and experiment design with section demonstrate the results.

2 Related Work

Recently literature pertaining to image classification in food domain has gained a lot of attention. Recent advancements in Deep Learning Convolution Networks has led to tremendous improvement in food based image classification. This paper proposes CNN based approach for food image recognition problem. Classification of images has garnered a lot of attention in past because of presence of lot of large scale datasets already present. **In this work we present a novel method to cluster these dishes(food items), which depends on their ingredients and thus is more organic, useful and has many more wide applications.**

Facebook has also worked extensively on Inverse Cooking which generates recipes given images. These is a large scale dataset that consists of 1M+ (million) training samples, which includes 1M+ images and their corresponding recipes. However none of these papers focuses on combining textual data along with image embeddings to learn optimal representation and image clustering to identify similar dishes

3 Objectives

We started with a project to identify ingredients by looking at images. Later at some point we realized that we can make it more useful and novel by using those embedding and as an application try to cluster dishes on basis of those. This was also an attempt to make it novel and doing something which was not attempted before. The problem statement becomes ever more challenging because

1. there are no direct large datasets of images with ingredients
2. In most cases ingredients provided are not clean and hence require a lot of preprocessing to even start with.

This approach helped to make more abstract ingredients and just using embeddings make it more useful. Blog

4 Resources Used

4.1 Datasets

This was the major dataset we used. This has around 13K images and their ingredients, but because these are not just ingredients (contains their measurements as well), we had to do a lot of preprocessing on the input data.

4.1.1 Preprocessing Steps

The dataset consists of textual and image information both of which require a different type of preprocessing

4.1.2 Images

1. All the images are augmented with different orientations(flipping images left,right, up and down).
2. Further images are added by changing the contrast, brightness and hue of the original image.Training the network on the original along with augmented images help the network learn optimal image representation for clarification.

4.1.3 Ingredients

We started with **13,533** ingredients because there were ingredients with different names for example "butter at room temperature" and "butter", different types of cheese is not different from each other for our use case. We quickly realized that this is not going to scale because of sparse nature of problem statement. We devised a set of rules to limit the number of distinct ingredients to equal **1872**.

1. Removing unnecessary special character and digits and converted all words into lowercase letter.
2. Removed measurement description given in ingredients list.
3. Tagged all the word tokens into part of speech and kept only noun one.
4. Created a set of negative words which was removed from the ingredient word list in case if its present.
5. reducing two ingredients to one in case they share the last two words.
6. reducing two ingredients to first or last word in case same word is present in first or last position.

These last two points are taken from Inverse Cooking paper.

After doing all these pre-processing steps, we convert all this data in TFRecords so that we can use them across platforms and for easy management.

4.2 Pretrained Models(Images)

Training a CNN from scratch has multifold issues

1. Requires a lot of data to train the network.
2. Very deep CNN's are computationally very expensive and requires a lot of time to train.

Transfer learning is popular method used in image classification domain which utilises a pretrained model to learn embeddings of the images in the current dataset. Pre trained model are trained on huge datasets and that help to learn optimal weights in the network, which can later be finetuned with relatively small datasets, which we do in this case.

We evaluated a lot of pretrained models to be able to identify the best on basis of our loss function. We used these different models with "imagenet" weights: EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6, EfficientNetB7, InceptionResNetV2, InceptionV3, ICPV2, ICPV3, XCP, VGG16, Resnet50.

4.3 Pretrained Models(Text)

Similar to images, in order to generate textual representation one uses pre-trained models trained on huge corpus of textual data. Glove is one such pretrained model which we evaluated in our work to represent ingredients' tokens. We compare them in later sections.

5 Methodology and Experiment Design

Our work is a combination of two models. First model is used to predict ingredients or in other sense the penultimate layer(of size 256) which is later used as the feature generation or embeddings layer. Second model is the model which takes the previous model as the encoding network and then trains a projection layer to predict features to be able to predict projection features. In second self supervised model, we do not train encoder network, to make sure that our projection is based only on ingredients.

To pick first model, we tried two different approaches. One was to only use images as training examples and other was using them with Glove Embeddings of the titles. We have summarized results for both of those in subsequent sections, but in a nutshell, results were better in case we only used images, so we pick those over the other. We save the weights of best model (in terms of loss) and then later use it for our subsequent modeling.

In our experiments we use distributed TPUs to facilitate faster training.

For second model (i.e. the clustering model) we use self supervised **Contrastive Loss** to cluster images along with **l2 normalization**.

5.1 Solution Architecture

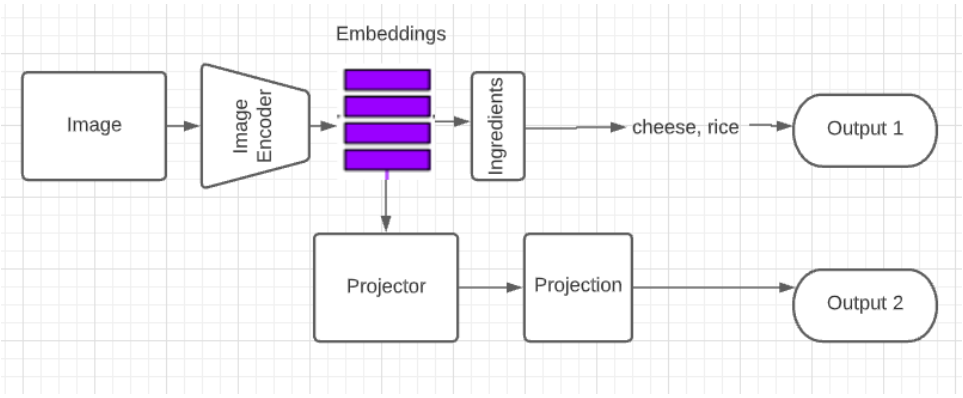

Figure 1: Solution Architecture

6 Results

We started with evaluating different pre-trained models to be used as image encoders, below is their loss on validation set.

After these experiments were performed, we made a decision to choose Resnet50 as our base model. We use these weights and trained a project to demonstrate the usability of our embeddings and results were good. Below are dishes from 4 different clusters with their ingredients included in next image. Though it is difficult to directly correlate images with ingredients (since this network does not include the weights of final layer), they look very close in terms of those.

Model	best loss (15 iterations)
eff1	0.0.02553
eff2	0.0.02479
eff3	0.0.02547
eff4	0.0.02962
eff5	0.0.02051
eff6	0.0.02440
eff7	0.0.02498
XCP	0.0.02027
RN50	0.0.01947
ICPV2	0.0.04324
VGG19	0.0.01973
VGG16	0.0.01978
RN101	0.0.01949

Table 1: Loss with images

Model	best loss (15 iterations)
eff1	0.0.02689
eff2	0.0.02496
eff3	0.0.02553
eff4	0.0.02362
eff5	0.0.02044
eff6	0.0.01980
eff7	0.0.01969
XCP	0.0.01954
RN50	0.0.02022

Table 2: Loss with 50D Glove Title embeddings + images

Figure 2: Clustering Results

```
['room', 'kosher', 'oil', 'flour', 'cream', 'cut', 'stick', 'sugar', 'tablespoon', 'vanilla', 'surface']
['butter', 'kosher', 'oil', 'flour', 'cut', 'rice', 'quinoa', 'buttermilk', 'pan', 'soda', 'wheat', 'sunflower'
, 'millet']
['pinch', 'oil', 'inch', 'sugar', 'syrup', 'half', 'vanilla', 'loaf', 'challah']
['ground', 'flour', 'brown', 'sugar', 'tablespoon', 'canola', 'sea', 'cornmeal', 'cold', 'rhubarb']
['room', 'salt', 'butter', 'kosher', 'flour', 'milk', 'baking', 'water', 'sugar', 'tablespoon', 'zest', 'vanill
a', 'pan', 'plain', 'rhubarb']
['flour', 'cream', 'sugar', 'rice', 'syrup', 'orange', 'tablespoon', 'vanilla', 'sea', 'bittersweet']
['lime', 'vodka', 'triple']
['water', 'lemon', 'sugar', 'pound', 'yeast', 'bottle']
['kosher', 'water', 'lime', 'plain']
['lime', 'ice', 'grapefruit', 'ounce']
['wine']
['rum', 'lime', 'syrup', 'garnish', 'ice']
['ground', 'butter', 'egg', 'milk', 'chopped', 'baking', 'tomato', 'bacon', 'sea', 'mozzarella', 'beaten']
['salt', 'butter', 'flour', 'powder', 'milk', 'lemon', 'sugar', 'vanilla', 'cheese']
['ground', 'salt', 'pinch', 'flour', 'water', 'sugar', 'pound', 'tablespoon', 'canola', 'vanilla', 'pineapple',
'soda', 'cane', 'pastry', 'organic']
['pinch', 'g', 'cornstarch', 'mustard', 'sauce', 'sea', 'beer', 'sel']
['cream', 'baking', 'stick', 'sugar', 'ginger']
['salt', 'flour', 'pepper', 'inch', 'milk', 'baking', 'lemon', 'mayonnaise', 'corn', 'tablespoon', 'food', 'bas
il', 'cold', 'pie']
['thyme', 'lemon', 'mayonnaise', 'paste', 'fillet', 'tablespoon', 'zest']
['onion', 'chicken', 'oil', 'chopped', 'pound', 'juice', 'arugula', 'accompaniment', 'vinegar', 'cremini']
['ground', 'salt', 'butter', 'flour', 'powder', 'milk', 'cream', 'baking', 'accompaniment', 'vanilla', 'mild']
```

Figure 3: Image wise Ingredients of Clustered Images

This last image shows the loss of validation set as we progress in our training.

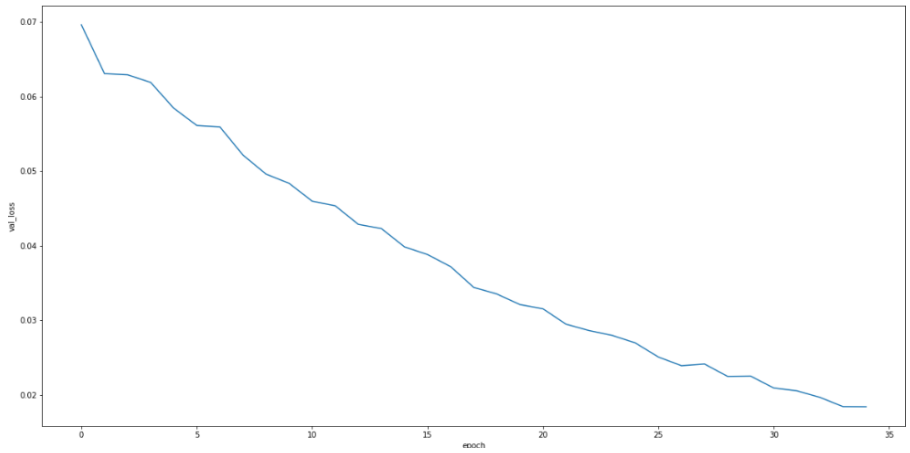

Figure 4: Validation Set Contrastive Loss

7 Conclusion

Even though classification of dishes has garnered a lot of attention in past in terms of which cuisine they belong to (probably because presence of lot of datasets) etc, we felt that there was an absence of novel work which tries to distinguish (or cluster) dishes in terms of dishes, even though the later might sound more useful and might find a lot more applications. This work tries to achieve same through a amalgamation of different techniques we learnt over the course. Following are the final outputs of our work.

1. Code repository
2. Cleaned and semi cleaned datasets
3. Final TFRecords

8 References

1M+ recipe dataset
Glove embeddings
Inverse Cooking
Example to create TF records
example for semantic clustering