Finding The Ingredients of Pizza Using Deep Learning

Mümin Can Yılmaz
can.yilmaz12@hacettepe.edu.tr

Alim Giray Aytar
giray.aytar12@hacettepe.edu.tr

Hayati İbiş
hayati.ibis12@hacettepe.edu.tr

Abstract

Extracting ingredients from a dish can be a powerful tool for combatting obesity and making food inspection processes easier. For this purpose, we tried to create a program which extracts ingredients from a pizza, using convolutional neural networks. We also created a dataset which has 7405 images and 20 different labels as ingredients. Our experiments show us our model can predict small numbers of ingredients successfully (80 percent for one label), however as the number of ingredients increased, accuracy rate drops significantly (22 percent for 2 labels).

1. Introduction

Our aim is to create a model which can identify ingredients in the pizza. Our program should output a list of ingredients as output when feed with an image of a pizza.

First of all, we started with creating a new dataset from the scratch, because we couldn't find any ready-to-use dataset. To do this, we collected about twenty five thousand images from web and labeled all of them by hand with a little software we created for this purpose.

Secondly, we decided to use a Convolutional Neural Network, because they show much better performance in image recognition problems compared to other approaches. Also when using Convolutional Neural Networks, we don't need to extract any features because CNN's operates directly on images. There is also some downsides of using Convolutional Neural Networks as they need more data and require more computing power than other solutions.

Finally, we evaluated our project with the result that we get after the process of training our classifier model which we present in the results section.

Hardest part of this problem is, because food shapes are deformed after cooking, it might not be possible to predict them correctly for our model. Color information also isn't very helpful, because some different ingredients exactly have the same colour or same ingredients might have different colours.

2. Related Work

In our research for related works, we focused especially on the projects with the food recognition and ingredient extraction problems.

In Jay Baxter's paper[3], they used a SVM (Support Vector Machine) as model their model. And their data set is The Pittsburgh Fast Food Image dataset (PFID)[4]. But, since they were not using Convolutional Neural Network, their features for SVM was important. Their work on features extraction depends on the clustering that they done on the training images. Thus, they were able to extract ingredients from clustering, then feed it into the SVM.

In Food Recognition Using Statistics of Pairwise Local Features paper [5], they take a pair wise local features as their starting point. But the model they used is not a network, but standard baseline algorithms specified by PFID(pairwise features): color histogram + SVM(Support Vector Machine) and bag of SIFT features + SVM.

In Deep-based Ingredient Recognition for Cooking Recipe Retrieval paper[6], their first goal was the ingredient recognition like our problem. But general problem is multiple label enabled in the prediction. DCNN models like AlexNet[7], with their preference of loss function, their model tries to boost the probability of one specific label. Thus, they divided their Neural Network into 4 different network, and trained them simultaneously. With multiple networks, updating the parameters done more freely for optimization of individual network performance.

3. Dataset

3.1. Research

Our dataset consists of raw images (their RGB values) and corresponding labels. Because we are using convolutional neural networks, we didn't have to extract any other features to use other than that.

At first, we started looking for a specific data set that could provide what we needed in the project. First dataset that we came across was the Food-101 dataset[1]. But we needed more specific dataset, as we only use pizza images. As we advanced our research, we saw that there was no dataset on the web that is ready for us to use or in any other academic projects which is similar that we could have used. Thus, we needed to create our own dataset from scratch.

3.2. Gathering Related Images

After deciding to build our own dataset, we started gathering pizza images from the Internet. For this purpose, we wrote a simple web crawler program with Python to download the needed pizza images from the corresponding web sites.

First, we started with web sites related to food or pizza. The pizza images in these related web sites were already labeled. But, the number of data samples that we could have get from these sites were too few for our project. Thus, we started to download images with tag of "pizza" and pizza related tags

from the Instagram[2] by our crawler. The problem with this was, the images that we gathered had no ingredient labels. Thus, we needed to label these images by hand.

For this reason, we created a web site which would help us with labeling process. The imageset we download from the Instagram[2] had some unrelated images and some of the related images had some noise (pizza was at the particular part of the image). With this web site, we were able to delete the unrelated images and crop the pizza out of images that included a pizza in some part of that image.

3.3. Creating The Dataset

In the end, we were able to gather 7405 samples with their corresponding labels. But, for training our Convolutional Neural Network we needed much more samples. Thus, we used the flipping, random cropping and some other techniques from the data augmentation methodology to increase the number of samples in our dataset. After the data augmentation process, we ended up with 148020 images and their corresponding labels.

Before the data augmentation process, we reorganized the labels. In total, we had 32 ingredients for every image. These 32 labels are; mozzarella, cheddar, tomato, pepperoni, onion, green pepper, salami, mushroom, meat, olive, chicken, basil, red pepper, corn, jalapeno, green olive, sausage, egg, parsley, cress, shrimp, potato, broccoli, BBQ sauce, rucola, lettuce, avocado, eggplant, tuna fish, pickle, lemon, carrot. But some of the labels were uncommon in the data set. These uncommon labels are ; shrimp, potato, broccoli, BBQ sauce, rucola, lettuce, avocado, eggplant, tuna fish, pickle, lemon, carrot. So, we reduced the number of labels from 32 to 20 by deleting the 12 most uncommon labels.

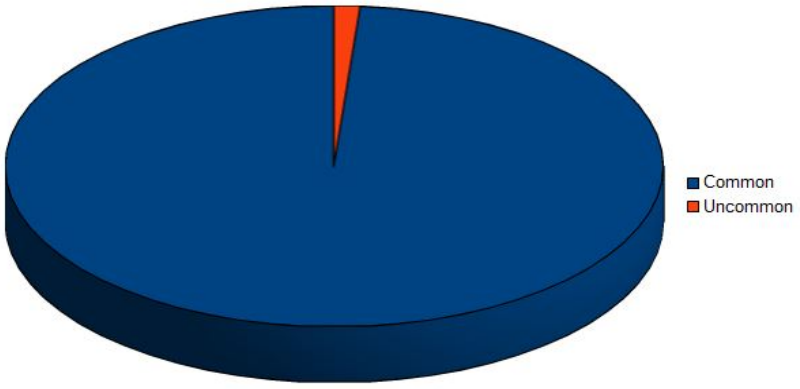

Figure 1. Data Set Ingredients Common-Uncommon Distribution

For computational reasons, we merged the RGB channel values and corresponding labels of our data set into a binary files and every binary file consist of a number of samples corresponds to our batch size.

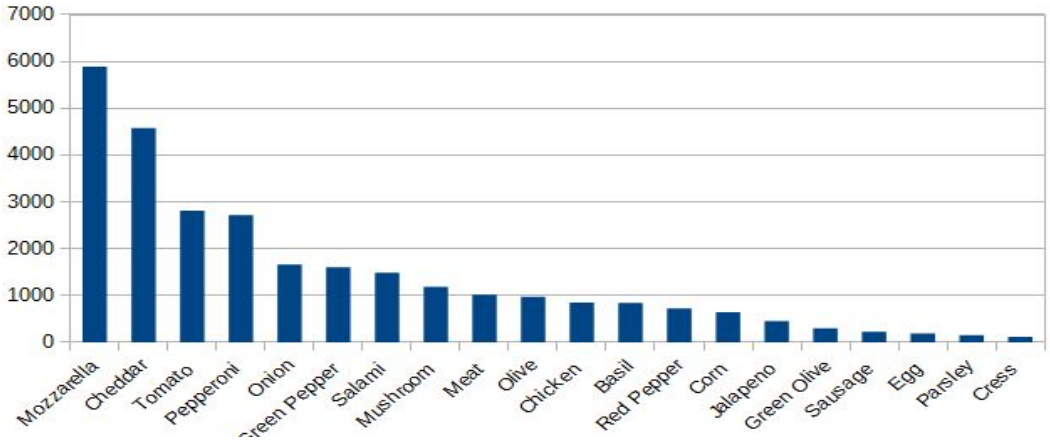

Figure 2. Data Set Common Ingredients Distribution

4. Convolutional Neural Network

4.1. Researches about CNN

We focused on AlexNet[7] paper.We learned new methods from this paper.We tried to use these methods as much as possible.(dropout,data augmentation etc.)We created our model by considering the common points of other models we have seen.

In detail we discovered an architecture called inception.After doing some research on this architecture, we decided that it is a very complex structure for our current problem.So, we didn't use this approach in our project.

4.2. Structure of Our CNN Model

We build a model which has three convolutional layers, two pooling layers and two fully connected layers.
Full Architecture is below:
[32x32x3] INPUT
[23x23x32] CONV1: 32 11x11 filters at stride 1, pad 1
[12x12x32] MAX POOL1: 2x2 filters at stride 2 pad 1
[9x9x64] CONV2: 64 5x5 filters at stride 1, pad 1
[5x5x64] MAX POOL2: 2x2 filters at stride 2 pad 1
[4x4x256] CONV3: 256 3x3 filters at stride 1, pad 1
[4096] FC4: 4096 neurons
[4096] FC5: 4096 neurons

We have some disadvantages in our model and It is related with our computational power problems.The disadvantage is starting from the first stage.We used [32x32x3] input, and if we used [64x64x3] or [128x128x3] instead, we could build a more successful model.(We can try them to get more successful results in the future.) Despite all this, the model in our case gave us good results.

We used ReLU (rectified linear unit) as activation function based on other successful research projects and the reason of ReLU doesn't saturates the neurons gradient and it doesn't slow the training down. When calculating the cost we used sigmoid cross entropy function instead of softmax. Because, softmax loss function tries to maximize the one-true label(one-hot label). We could have used a loss function which tries to minimize over-all histogram of predicted labels.But, the used framework was not included that type of loss function. Multi-Label classification is still a research topic. We considered training multiple classifier with changing our model. But, for computational reasons we chose this model.

4.3. Training Our CNN Model

There are many frameworks we can use it (Caffe,Torch,Theano,Minerva,CXXNet,TensorFlow). They all have some advantages and disadvantages.We prefer TensorFlow.It is the newest released framework about machine learning.

In the training step, we need to determine about parameters; learning rate, batch size and dropout.

We set them:

Parameter	Value
Learning rate	0.001
Batch size	100
Dropout	0.50

Firstly, we tried to use [256x256x3] and [128x128x3] sized inputs.They took immeasurable run times, then we decided to use [32x32x3] inputs and It's execution time took 2800 seconds.

5. Experimental Results and Evaluations

5.1. Results

Our experiments and results are as follows:

Labels	Accuracy
1	0.80
2	0.22
20	0.01

Labels means that our model predicted at least n numbers of labels correctly. If n=1 it means that at least one label is correctly predicted, if n=2 it means that at least two labels are correctly predicted and so on.

For some reason our model's accuracy on predicting all labels correctly is fairly low. It seems to predicting always more labels than originally existed. For example, when we consider an image with four ingredients, the prediction says there is five ingredients in this image, which also contains the original four ingredients, but the result becames false because the array's didn't match.

5.2. Evaluations

Our CNN model is good at predicting an ingredient is appear in the image or not, but it is not very successful predicting exactly which ingredients are in the image. This may be happening because some ingredients are very similiar to each other or there is some noise in our datasel labels. Also our model can't distinguish different looking ingredients, such as green tomato and red tomato.

6. Conclusion

Food recognition and ingredient extraction are new but growing research areas. It is also a little bit more difficult to solve compared to general image classification problems because of the unique situations related to this problem, such as object deformation, invisible ingredients or using different cooking methods.

Invisible ingredients include olive oil like ingredients which are used, but not visible to eye. There is also some ingredients which also exists in pizza but invisible in photograph. We currently don't know how they affected the result of our experiments.

From our experiments we see that using solely Convolutional Neural Networks might not be the best way to solve this problem, because there is a lot of extra information we can use. For example, we know some ingredients are used or not used together, also we know that same ingredients may appear in different shapes and colours. These problems can be solved using more advanved models and also using bigger datasets.

In future work, we plan to use a bigger and more refined dataset. Also we are planning to extend our model to increase it's accuracy and developing practical food recognition applications.

References

[1] Lukas Bossard, Matthieu Guillaumin, Luc Van Gool, " Food-101 – Mining Discriminative Components with Random Forests"
http://www.vision.ee.ethz.ch/datasets_extra/food-101/

[2] Images from Instagram with tag of "pizza" https://www.instagram.com/explore/tags/pizza/

[3] Food Recognition using Ingredient Level Features Jay Baxter - MIT http://jaybaxter.net/6869_food_project.pdf

[4] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In Image Processing (ICIP), 2009 16th IEEE International Conference on, pages 289–292. IEEE, 2009.

[5] Food Recognition Using Statistics of Pairwise Local Features http://homes.cs.washington.edu/~shapiro/cvpr10.pdf

[6] Deep-based Ingredient Recognition for Cooking Recipe Retrieval http://vireo.cs.cityu.edu.hk/papers/jjchen_mm16_cr.pdf

[7] ImageNet Classification with Deep Convolutional Neural Networks https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networ pdf