DishClose – Recipe Retrieval App

Vajid Kagdi, Rohan Acharya, Sravya Somisetty, Heli Desai

Software Engineering, San Jose State University
1 Washington Sq., San Jose, CA 95192

vajid.kagdi@sjsu.edu
rohan.acharya@sjsu.edu
sravya.somisetty@sjsu.edu
helidipakkumar.desai@sjsu.edu

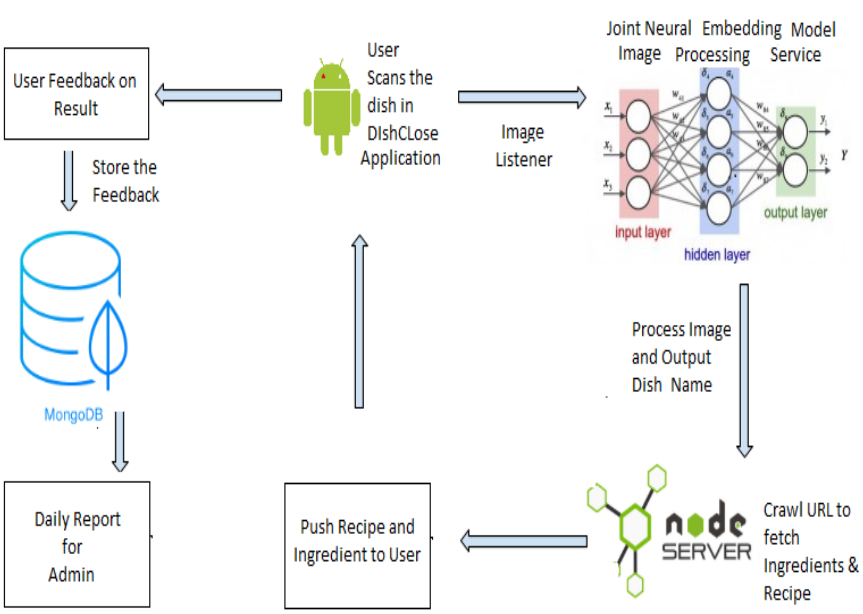
Abstract— **This paper describes solution implemented for common people who want to know the recipe and/or ingredients of a food dish. User can scan pictures on the DishClose mobile application, which will then generate an automated list of ingredients and recipes using image recognition algorithms on past data. The application will also take user feedback in the form of 'Yes/No' responses to the queried information. It will use this to further improve its recognition accuracy.**

***Keywords*— residual neural network, stacked non-linear layers, image recognition, probability threshold, confusion matrix, precision, recall, San Jose state University**

I. Introduction

This document is a final report from our CMPE 272, Spring 2018 Enterprise Software Platform as graduate students. Our mentor, Prof Rakesh Ranjan directed our effort in this project for providing user-centric solution and eventually, we came up with this android application which can help customers to know the recipe/ingredients of their favorite dishes.

A. Architecture:



1) 1) Android Application:

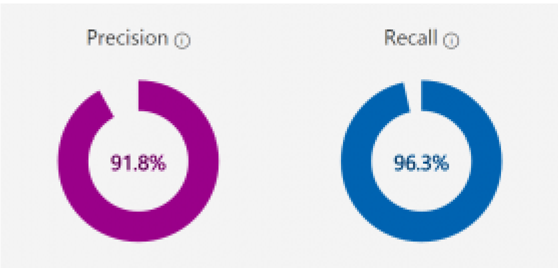
We have created a smart android application in which a user can scan the image or the actual dish and the application will recognize the dish. Unlike other application where user needs to take a picture and upload it, we have simplified the process by adding a Image Listener in android which continuously check the current frame and displays the dish name if it recognizes with accuracy greater than 70%.This makes the process faster. Once the dish name is recognized, user can tap on screen to fetch the ingredients and recipe of the same. We have used Microsoft Custom Vision Service for the image recognition part which provides us with the dish name. Once the dish name is recognized we have created a REST API using Node JS which takes input the name of the dish and finds the best rated recipe as per Food2Fork ratings. Food2Fork provides the URL of the recipe. Once the recipe URL is available from Food2Fork service, the next step is to crawl the URL to fetch the ingredients and recipe form the URL. In response, the Node JS API provides JSON object containing list of ingredient and steps to cook. Users here get the option to provide the feedback which will be stored in MongoDB using Node JS API. The application also remembers the history of the scans till date and user gets the facility to fetch the recipe later. In this way it becomes a convenient option for tourists and people who migrate to learn and get accustomed to new dishes.

For the admin part, we have provided a daily feedback report facility which will provide the report of user feedbacks till date in the form of Email. With the help of this report, admin will be able to analyze the false positive and false negative detections and thereby can take steps to improve the model by improving the training dataset.

2) Image Recognition:

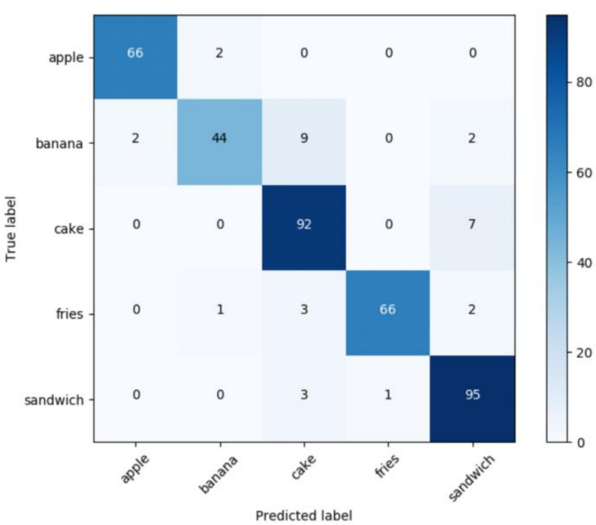
We have used Microsoft Azure's Custom Service Vision API for food dish recognition. The algorithm is based on the concept of Residual learning, which means each subsequent layer in a deep neural network is only responsible for, in effect, fine tuning the output from a previous layer by just adding a learned "residual" to the input.

Once the model is trained, the statistics below are displayed in the Custom Vision portal for each iteration. We had pretty good model performance out of the box with both precision and recall exceeding 90%! Precision metrics tell us the percentage of correct predictions for a given image. Recall measures how much a classifier can detect (what percentage of the apples in the test were classified as such).



Custom Vision introduces a “Probability Threshold” slider (we used the default value of 90%) that is used to calculate Precision and Recall. When interpreting the predictions you get a probability per tag; for example, the probability that picture A contains an apple is 95%. If the probability threshold is 90%, then this example will be taken into consideration as a “correct prediction”. Depending on your application needs, you may want to set a higher/lower probability threshold.

Custom Vision uses something called a Confusion Matrix; items on the diagonal are cases where the model’s prediction is correct (that is, 95 images were predicted to be sandwiches and they were in fact sandwiches). In our model, train and test images contain one type of food and when interpreting Custom Vision predictions and building Confusion Matrix we focus on the one with the highest probability.



The numbers that are off the main diagonal show where the model made classification mistakes. Nine bananas were incorrectly classified as cakes, and seven cakes as sandwiches.

Those mistakes give us good insight that the color and the texture-rich variety of cakes make it quite complex for the model to learn the requisite class features; as a result, this class is easy to confuse with something else.

	precision	recall	f1-score	support
apple	0.97	0.97	0.97	68
banana	0.94	0.77	0.85	57
cake	0.86	0.93	0.89	99
fries	0.99	0.92	0.95	72
sandwich	0.90	0.96	0.93	99
avg / total	0.92	0.92	0.92	395

B. Application in Action:

1) *User scans over the food dish.*
2) *The application throws an expandable pop-up with the ingredients.*
3) *The application also provides shortened recipe steps, which can be expanded by hovering over the text.*
4) *User is the asked for feedback about the accuracy of the results.*
5) *User can access the History page to check up on the most recent scan results.*

II. Conclusions

.

At first, out research experience on dividing problem space, designing application architecture we found many problems like which machine learning model should we use? Which storage database would be good for storing training and testing images and storing user feedback? While working on machine learning techniques, we found the residual neural network (ResNet) giving us the best results. We learnt that fine-tuning end-to-end on a pre-trained network with relatively less volume of data can still give us high accuracies.

Acknowledgment

Thanks to Professor Rakesh Ranjan for his impeccable and timely guidance and the motivation he provided to go beyond expectations. Also, a big thank you to the Custom Vision Service team from Microsoft Azure, using whose image recognition API, we could build this application.

References

[1] Microsoft’s image recognition service- https://www.microsoft.com/developerblog/2017/05/12/food-classification-custom-vision-service/
[2] Deep Residual Learning for Image Recognition - https://arxiv.org/pdf/1512.03385.pdf
[3] Food2Fork: http://food2fork.com/about/api
MShape:https://market.mashape.com/spoonacular/recipe-food-nutrition#extract-recipe-from-website
[4] Daily Admin Email Report: https://nodemailer.com/about/
[5] MongoDB Deployment: https://mlab.com/
[6] Node Js: https://nodejs.org/en/