Food Recognition Software for a Web-App

Team - Alcove

Heena Khan

Luis Chunga

Steven Sheffey

James Phillips

Matthew Radice

Elijah Barbour

Mason Thieman

Introduction

- This project uses convolutional neural network to generate a food recognition model for a **Free-Food Finder** webapp.

- The model is trained on a very famous dataset **Food-101** provided by Kaggle and originally from the research paper "Food-101 – Mining Discriminative Components with Random Forests" by Lukas Bossard.

- This is challenging dataset consisting of **101,000 images** of **101 different food classes**. On purpose, the training images were not cleaned, and thus still contain some amount of **noise**. This comes mostly in the form of **intense colors** and sometimes **wrong label.**

Dataset

This is how images in our dataset looks like. Even as a human, I think I'd struggle to classify them.

Bread Pudding

Chicken Curry

Apple Pie

Cheesecake

Motivation

- Consumers **waste** approximately **41 million metric tons of** food per year [1].

- A proposed solution to this problem is a **Free-Food Finding app** developed by students at Middle Tennessee State University, which enables normal people to donate food which would otherwise go to waste.

- But what makes a user to like and use an app regularly?
  - It should be **User Friendly**
  - It should take **Less Effort**

- To make this app **effortless and user friendly** we propose the use of a Neural Network (NN) to automatically classify pictures of donated food that can reduce human effort.

Methods

- Model Architecture
  - NASNetMobile
- Dataset augmentation
  - Random rotations and shifts
- K-Fold cross validation
  - Validates model's ability to learn the dataset
- Metrics
  - Top 1,5,10 accuracy
  - F1-score

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
NASNet (Model)               (None, 7, 7, 1056)        4269716
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1056)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 1056)              0
_________________________________________________________________
dense_1 (Dense)              (None, 101)               106757
=================================================================
Total params: 4,376,473
Trainable params: 106,757
Non-trainable params: 4,269,716
_________________________________________________________________
```

Results

1. We have created 3
2. Model-1 is trained on entire dataset, Model-2 is trained on 50 random classes and Model-3 is trained on 25 random classes
3. We have calculated Top1, Top5 and Top10 predictions accuracies for all three models.

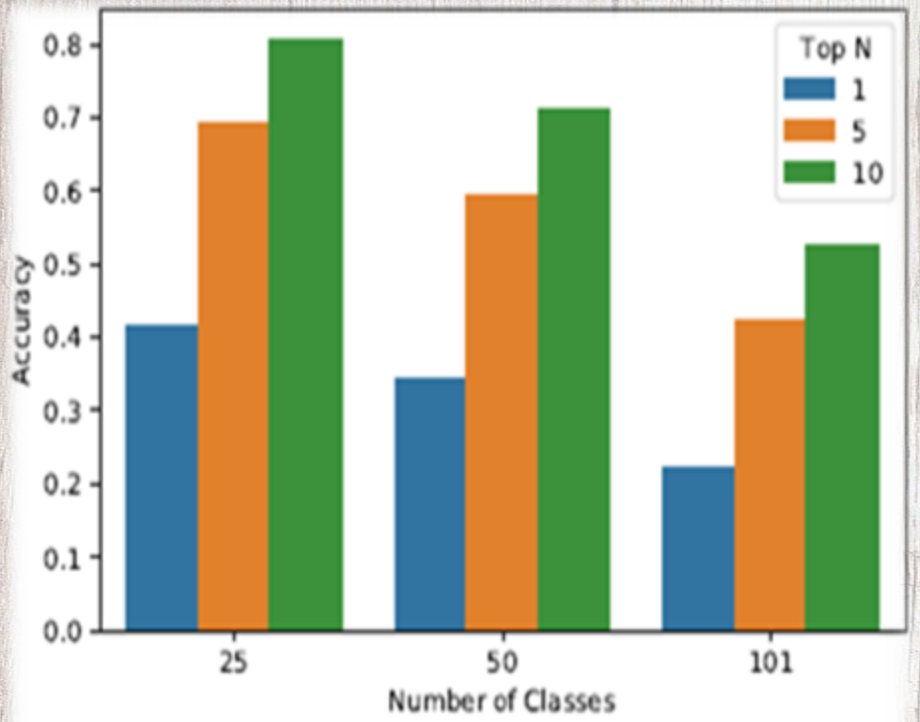

Fig. 2. Top N accuracy values across all 3 models

TABLE I
ACCURACY OF OUR MODELS, AVERAGED OVER 5 FOLDS

	Classes	Top 1	Top 5	Top 10
Model 1	101	22.23%	42.31%	52.56%
Model 2	50	34.30%	59.52%	70.99%
Model 3	25	41.37%	69.17%	80.68%

Our Contributions

James

Research and Demo Creation

- Research Gathering for Project
- Abstract, Introduction, and Background sections in paper
- Milestone
- Provided Documentation for the Demo

Elijah

Project Paper Documentation

- Mathematical Execution of Paper
- Research Gathering for Project
- Milestone

Luis

Research Methods

- Implemented a model with MobileNet for small dataset.
  - 5 classes
- Implemented a model with MobileNet With entire dataset.
  - 101 classes
- Create Interactive classification.
  - Display top 1 and 5 prediction
- Wrote the Model Architecture in the method section of the paper.

Steven

Experimental Design and Engineering

- Designed experiment around K-fold cross-validation
- Configured lazy data loading
  - Dataset augmentation
  - K-fold support
- Figure generation
- Paper review

Heena

Model visualization , Web app creation, Documentation and Team Management

- Found **Dataset** and came up with overall aims
- Created **Project Proposal**
- Implemented **Top5 and Top10** accuracies
- Generated **visualization** like **confusion matrix** and **accuracy plot** for Top1-5-10.
- Created the **Web app**
- Wrote **Methods** and **Result** section of our paper
- Created the **demo** walkthrough and the **Readme** page
- Co-ordinated team meetings and Initiated work and strategies

Mason

Project presentation and Documentation

- Presentation
  - Condensed relevant information from research paper into presentable slides, as well as creating individual contribution slides.
- Milestone Writing

Matthew

Research and Documentation

- Failed Miserably at contributing to network design

- Researched network design
  - NASNet – Designing a search space so the complexity of the architecture is independent of the depth of the network and size of the input images

  - Convolution Cells – This approach used convolution networks composed of 'cells' with identical structures but with different weights.
    - This is much faster than searching the entire network architectures.
    - The cell is more likely to generalize to other problems – good for our project.

  - The original NASNet was trained on a smaller image data-set, then using the best convolution cells trained on a larger image data-set on ImageNet.

  - Our network used the weights that were pre-trained on this ImageNet.

- I now understand there is a set of weights that could achieve 100% accuracy on this task and that I will not be the one to find these weights.

Let's look at our DEMO