Machine Learning Engineer Nanodegree

Capstone Proposal

Van Nhat Huy Phan

June 27th, 2018

Domain Background

Many machine learning techniques have been used everyday to improve our lives. However, usually new machine learning projects require a computer to run and deploy. This is not always the case because not everyone has a computer on them all the time. I want to solve a real life project, and have the solution available on a smartphone, a device we always carry with us. To solve these kind of problems, there 2 two main approaches: use the smartphones as a sending/receiving devices and put the trained model on a computer, or put everything on the smartphones including the trained model.

Problem Statement

When we hear a song and want to know its title and artist, we can use Shazam or SoundHound to find out. However, when it comes to food, there is no known mobile solution to recognize the dish. I want to create an app that allows us to recognize the dish just by pointing the camera at it. The app will take a picture of the dish, compare it will the trained model of 101 dishes, output the label with the highest score. This will be a classification task.

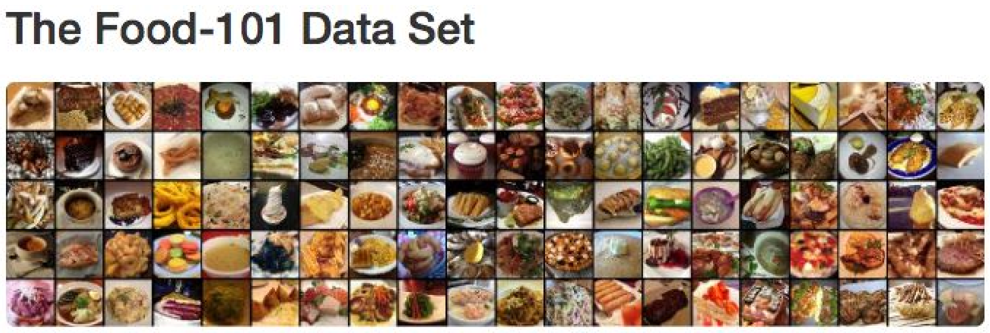
The app should help user to identify the dish the we could display see relevant information about the dish like nutrition and recipe.

The difficult part of this problem is that it quite difficult to put a deep learning model on a mobile phone which has little computing power.

Datasets and Inputs

I will use the public ‘Food 101’ dataset to train my model. ‘Food 101’ is data set of 101 food categories, with 101'000 images. For each class, 250 manually reviewed test images are provided as well as 750 training images. 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 labels. All images were rescaled to have a maximum side length of 512 pixels.

https://www.vision.ee.ethz.ch/datasets_extra/food-101/

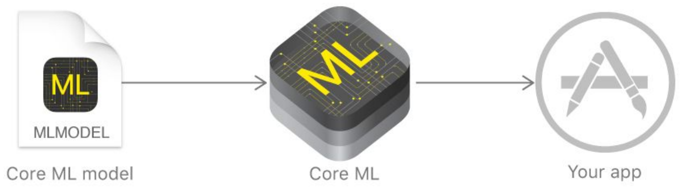


Benchmark Model

I will use AlexNet which is convolutional neural network (designed by Alex Krizhevs) as a benchmark model for this project. A convolutional neural network is perfect for this project because of its speed, ease to use and performance.
https://en.wikipedia.org/wiki/AlexNet

Solution Statement

With the release of CoreML by Apple, it is now possible to convert a trained model to a compatible format to use on iOS Apps.
I will use a convolutional neural network to train my model on the Food-101 dataset on an Amazon EC2 instance then convert to a CoreML to use on my iOS App.



Evaluation Metrics

- Since the classes are not skewed. Accuracy and Log-loss is quite enough to evaluate the model.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

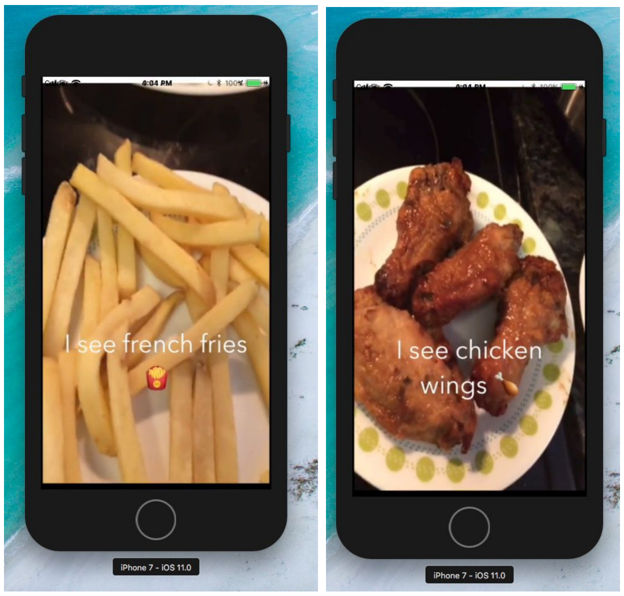
- The query time: time to recognize a dish. Should not be more than 3 seconds.
- Real life testing accuracy: the accuracy should be at least 90%. I will use different pictures on the internet as well as real life pictures to evaluate the accuracy.

Project Design

- My plan to implement the app consists of 3 parts: train, convert, and integrate
- Train: I will use Amazon Web Service to rent an instance to train the model with the Food 101 data set
- Convert: Once I have a trained model, I need to convert it to the CoreML model to use on iOS devices such as iPhone and iPad.
- Integrate: Set up a iOS app with simple interface to use the model.
- Steps to implement the project:
    - ❖ Set up AWS instance, install all required software and libraries.
    - ❖ Train the model in AWS using deep learning.
    - ❖ Convert the model to CoreML model, integrate to iOS app.

    - ❖ Evaluate, fix bugs, and complete the software.

- I intend to use ‘Caffe’ a deep learning framework made with expression, speed, and modularity in mind and Core ML Apple's machine learning framework. It can integrate machine learning models into my app.



The prototype app should consist of a camera view with dish information on top of it: