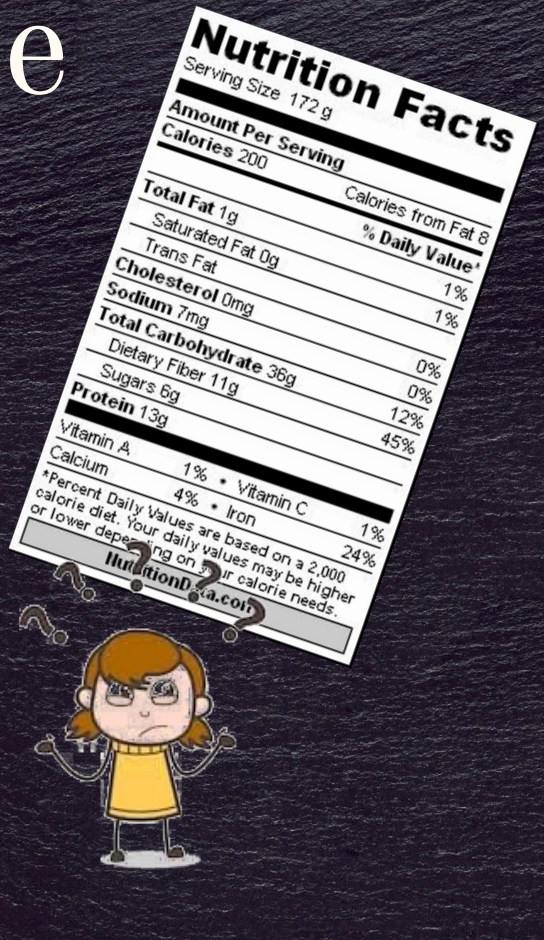
JALPAIGURI
GOVERNMENT ENGINEERING COLLEGE

Food Recognition & Nutrition Estimation using Deep Learning

Supervised: Prof. Swalpa Kumar Roy

Team: Sudipta Dandapat(16-4036)
Sudipta Sahana(16-4031)
Sanatan Nandi(16-4033)
Swarnakesar Mukherjee(17-4084)

How can we estimate the nutritional content of food just by looking at its image?

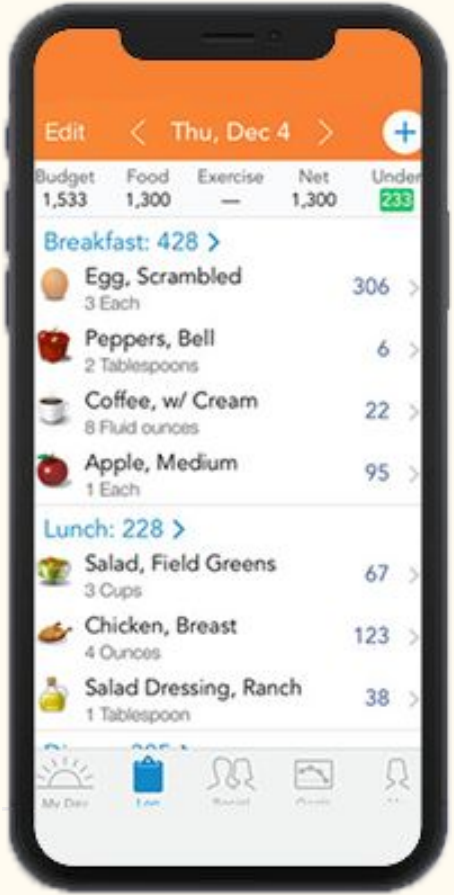


CONTENT

Introduction

Maintaining a healthy diet is an important goal for all people. Nowadays, fast food consumption has increased more and more and has became a daily diet of every individual. Fast food may be an easy option for us when we're in a rush, but its nutritional content definitely will not provide us the energy that we need for the rest of the day. So, it's important to estimate the nutritional content present in the fast food we eat.

We already have many different types of tools available online for Nutrition estimation, but, they assume that the user will enter some information about the food item consumed. For example, it might be expected that the user will enter the name of the food item or the ingredients, as well as the size of the food item and then run it against a static database of food items to be able to calculate the amount of calories in the user's consumed food item.



In this project, we came with an approach to alleviate the user to enter all such details and have the same result with **just a food image.**

We propose a **deep-learning based approach** to calculate the calories from the food image through classification of type of food and estimating the weight of the food. Here, we use a pipeline approach by doing few steps. First, we identify the type of food in the image. Second, we generate an estimated size of food item in grams. Then, by taking the first two intermediate results, we estimate different nutritional content from data-set.

Data Collection

Our dataset is based on the **Food-101** Data Set, which consists of images of food items belonging to over 101 categories. For each class, 250 manually reviewed test images are provided as well as 750 training images. In our dataset, we sampled 1,400 images from the Food-101 dataset, which were evenly distributed among 14 food types, namely: **Muffins, Noodles, burger, Egg roll, cheese pizza, pepperoni pizza, french fries, sandwich, hot-dog, fried chicken, biscuits, shish kebab, doughnut and tacos**. We restricted the number of images used and the food types to make it less time-consuming process, which had to be done very carefully in order to produce high-quality ground truth data.

And for the information of Nutritional content of these foods we use another dataset which is **Nutritional Data for Fast Food 2017**.

Item	Type	Serving Size (g)	Calories	Total Fat (g)	Saturated Fat (g)	Trans Fat (g)	Sodium (mg)	Carbs (g)	Sugars (g)	Protein (g)
Hamburger	Burger	98	240	8	3	0	480	32	6	12
Cheeseburger	Burger	113	290	11	5	0.5	680	33	7	15
Big Mac	Burger	211	530	27	10	1	960	47	9	24
Quarter Pounder with Cheese	Burger	202	520	26	12	1.5	1100	41	10	30
Bacon Clubhouse Burger	Burger	270	720	40	15	1.5	1470	51	14	39
Double Quarter Pounder with Cheese	Burger	283	750	43	19	2.5	1280	42	10	48
Chocolate Shake (12oz)	Milkshake	257	530	15	10	1	160	86	63	11
Premium Crispy Chicken Classic Sandwich	Breaded Chicken Sandwich	213	510	22	3.5	0	990	55	10	24
Premium Grilled Chicken Classic Sandwich	Grilled Chicken Sandwich	200	350	9	2	0	820	42	8	28
Chicken McNuggets® (4 piece)	Chicken Nuggets	65	190	12	2	0	360	12	0	9
Small French Fries	French Fries	75	230	11	1.5	0	130	30	0	2

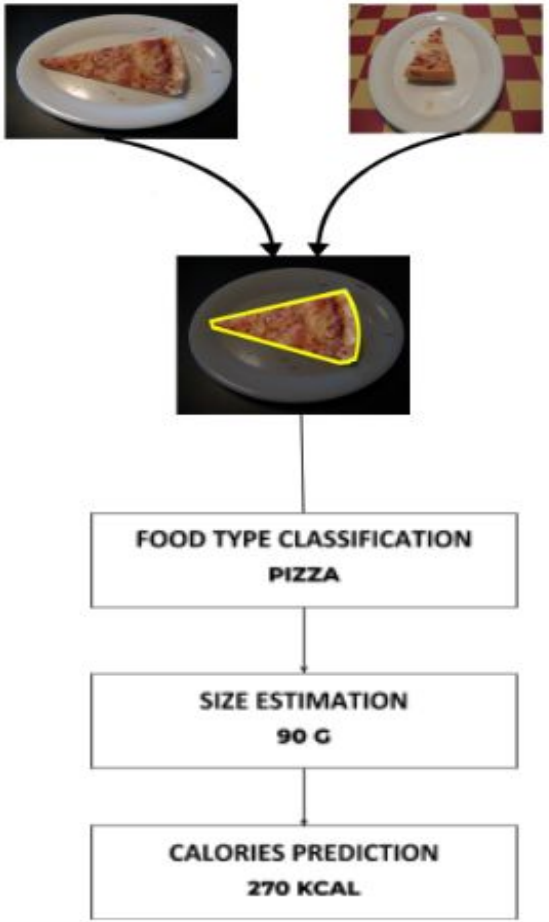
Our Methodology

Our system will take an input image of a food item and outputs the nutritional content present in this food item like **total fat, calories, carbs, sugar and protein**. To be able to do this we perform the following:

1. The image is passed through a classifier which classify the category of fast food it belongs to.

2. Next, we will do the volume estimation using a calibration object for the calculation of calories and other nutritional content.

3. Finally, the predicted type and size of the food item are passed to another regressor that predicts the amount of nutritional content in the food item.

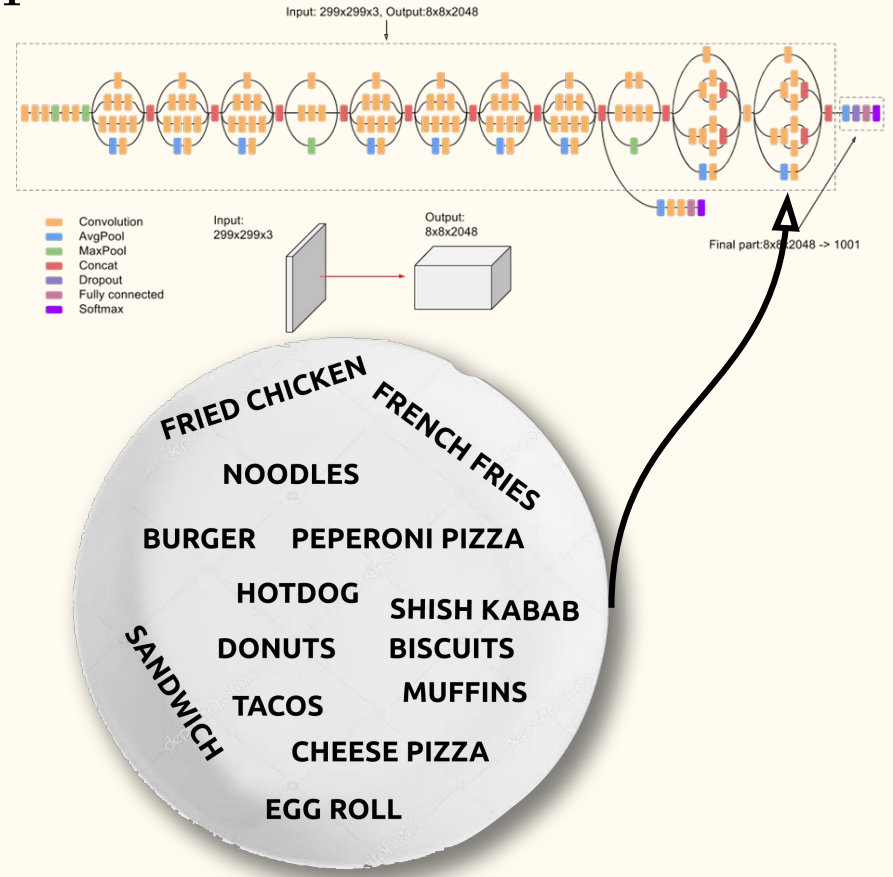


1. Food Type Classification

Given an image of a food item, our goal here is to predict the type of the food item.

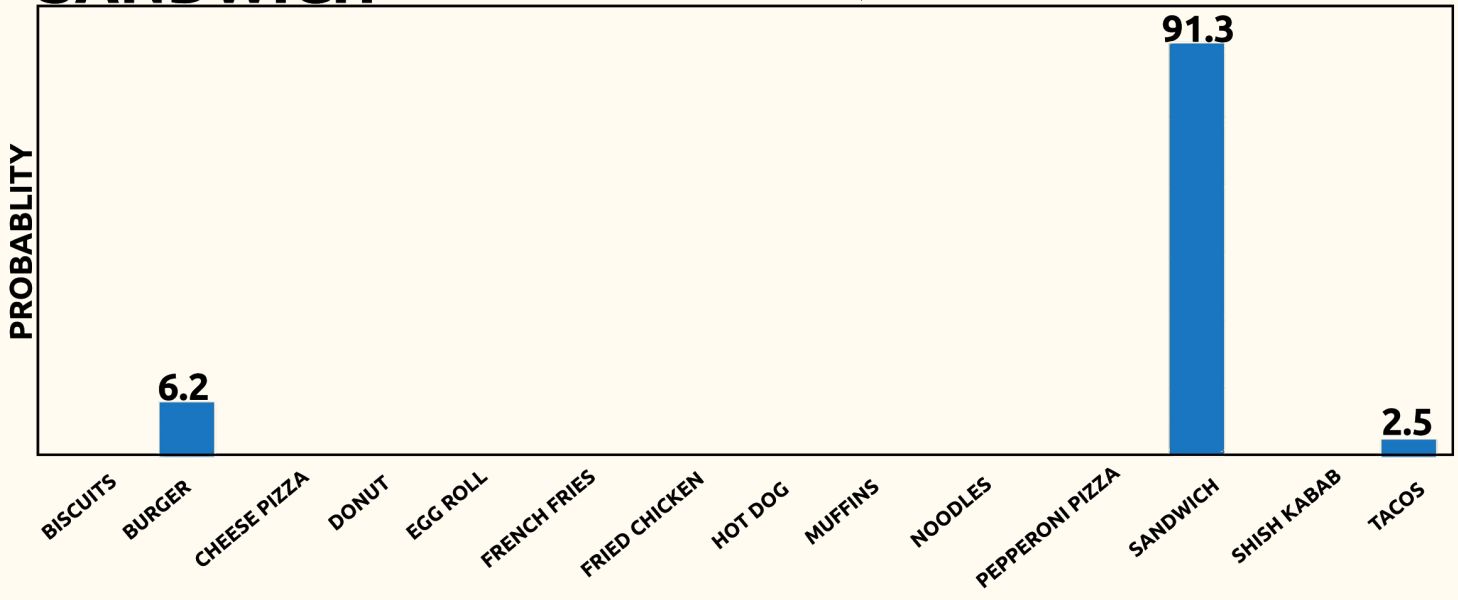
i. For this task we use the concept of Transfer Learning in which a pre-trained model is used to classify new images. In our case, we use the InceptionV3 model, which is the third iteration of the inception architecture, first developed for the GoogLeNet model.

ii. We then pass the feature extracted from InceptionV3 model to a classifier that outputs one of the 14 classes we are concerned with here. This becomes an additional feature that is fed to the Nutrition predictor that we will describe later.



FOOD TYPE CLASSIFIER

SANDWICH



2. Food Size Estimation

In this step, we need two input images from top view and side view to estimate the volume of the food. And each image should include the calibration object which is a 10 Rs coin in our case (diameter 2.7cm). For doing this, we will perform the following steps:

1. **Deep Learning Based Objection Detection:** We use YOLOv3 model to perform object localization and detection on images and draw boundary box around the main food object.

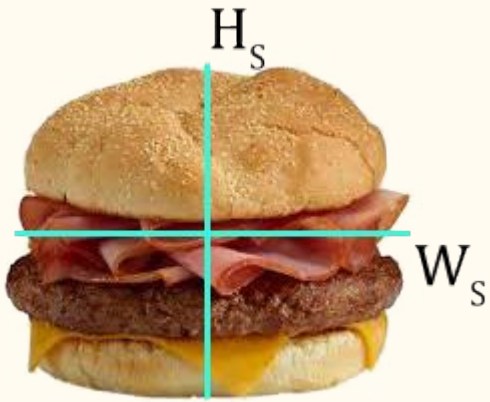
2. **Removing background and unwanted noise:** After segmentation of each boundary box, we will replace the values of the background pixels being by zeros. This will remove the unwanted background and leave only the foreground pixels.

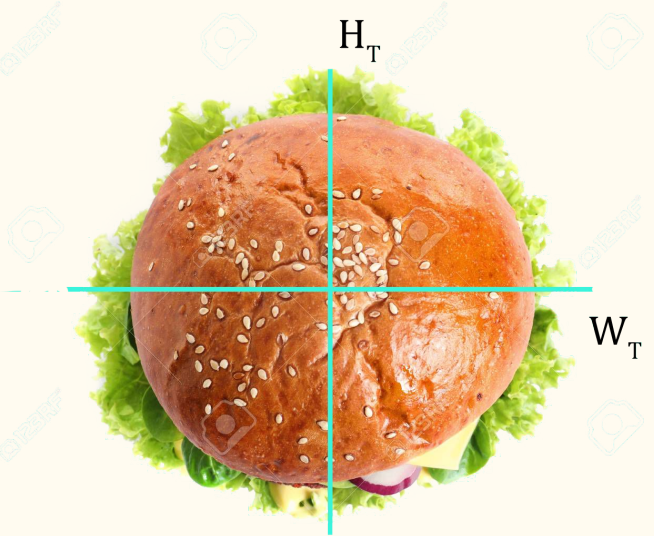
3. **Volume Estimation:**

To estimate the volume, we calculate the scale factors based on calibration objects(10 Rs coin of diameter 2.7cm). The side views scale factor $\boldsymbol{\alpha_S}$ and top views scale factor $\boldsymbol{\alpha_T}$ was calculated with Equation 1 and Equation 2 respectively.

$$\alpha_S = \frac{2.7}{(W_S + H_S)/2} \quad (1)$$

$$\alpha_T = \frac{2.7}{(W_T + H_T)/2} \quad (2)$$





Now, we estimate the volume using the following formula in Equation 3:

$$v = \beta \times (s_T + \alpha_T^2) \times \sum_{k=1}^{H_S} \left(\frac{L_S^k}{L_S^{MAX}}\right)^2 \times \alpha_S \quad (3)$$

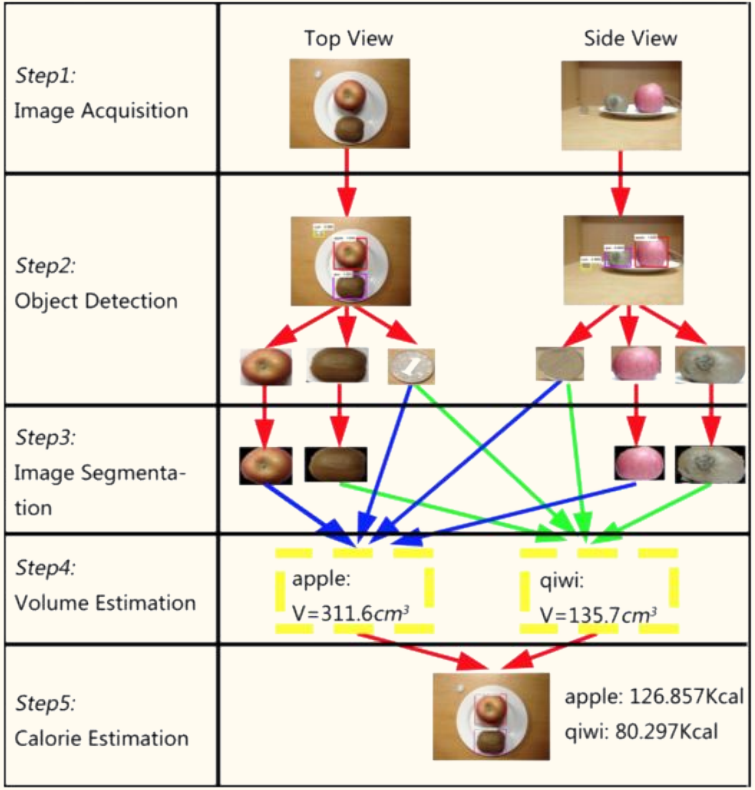
HS is the height of side view PS and LkS is the number of foreground pixels in row k ($k \in 1,2,...,HS$). $LMAX = \max(Lk,...,Lk)$, it records the maximum number of foreground pixels in PS. ß is a compensation factor (default value = 1.0).

4. **Mass Estimation:**

After estimating the volume, the next step is to estimate each food's mass. It can be calculated in Equation 4, Where v (cm^3) represents the volume of current food, and ρ (g/cm^3) represents its density value

$$m = \rho \times v \quad (4)$$

Flow chart of Volume Estimation to calculate Calories



3. Nutritional content Estimation

Finally, we describe our main task, which is Nutrition content estimation of a food item.

1. Given the input image we perform food type classification and size estimation and after that it passed to a regressor which outputs predicted amount of nutritional content in the food item.

2. Our regressor is trained with the **Nutritional Data for Fast Food 2017** dataset. This dataset contains information about Nutritional contents like **calories, Fats, proteins, carbs** and **sugar** in grams (g) of different types of fast food along with the serving size. Once the regressor is trained with the dataset, it will be used to predict the amount of Nutritional content in a food item.

NUTRITIONAL CONTENT ESTIMATOR

Nutrition Facts

Sandwich

1 Serving

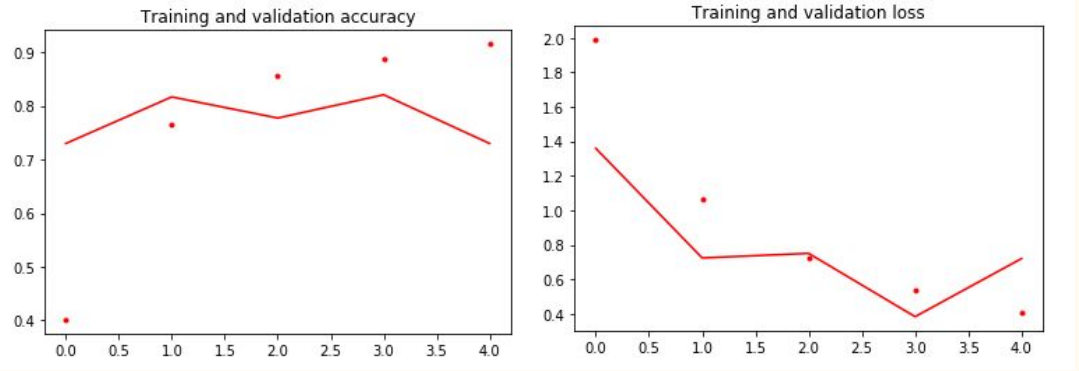
Amount Per Serving	
Calories	317.2
Total Fat	10.9 g
Sodium	277.4 mg
Total Carbohydrate	43.8 g
Sugars	0.5 g
Protein	13.2 g

Experimental Results

We present the experimental results food type classification:

i. After training the model with our dataset, we get Training Accuracy of **91.55%** and Validation Accuracy of **72.99%**.

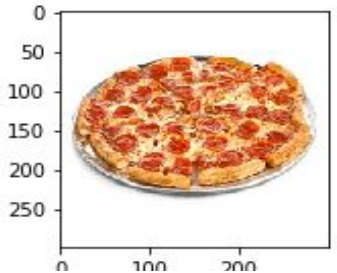
ii. We had shown the plots for Training Accuracy vs Validation Accuracy and Training loss vs Validation loss.

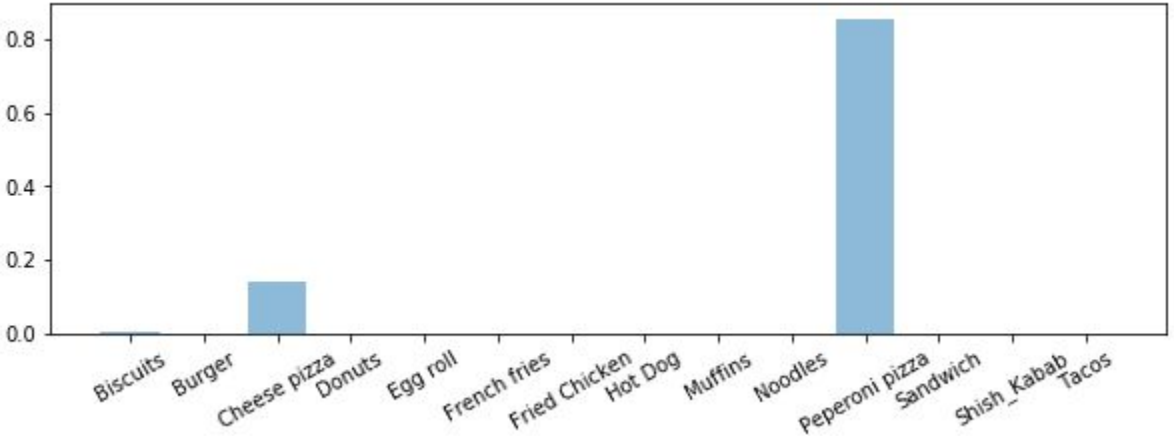



```
Epoch 1/5
50/50 [==============================] - 781s 16s/step - loss: 2.0214 - accuracy: 0.4014 - val_loss: 1.3593 - val_accuracy: 0.7
299
Epoch 2/5
50/50 [==============================] - 720s 14s/step - loss: 1.1284 - accuracy: 0.7656 - val_loss: 0.7234 - val_accuracy: 0.8
167
Epoch 3/5
50/50 [==============================] - 703s 14s/step - loss: 0.7846 - accuracy: 0.8557 - val_loss: 0.7500 - val_accuracy: 0.7
774
Epoch 4/5
50/50 [==============================] - 736s 15s/step - loss: 0.6294 - accuracy: 0.8880 - val_loss: 0.3827 - val_accuracy: 0.8
207
Epoch 5/5
50/50 [==============================] - 547s 11s/step - loss: 0.4684 - accuracy: 0.9155 - val_loss: 0.7198 - val_accuracy: 0.7
299
```

Output of Food Type Classifier

```
Out[54]: array([2.6111677e-03, 8.1516548e-05, 1.4271200e-01, 1.8771812e-05,
                1.3164656e-04, 1.1512225e-05, 3.9308568e-05, 1.8925935e-05,
                1.6672822e-04, 1.4358731e-04, 8.5369164e-01, 1.1417697e-04,
                6.7566441e-05, 1.9141388e-04], dtype=float32)
```



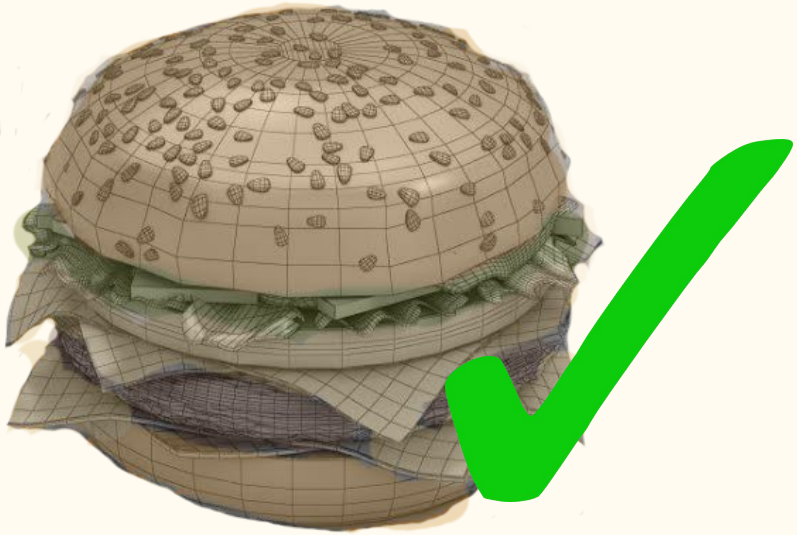


Future Scope

In future, we will extend our work with the following:

i. We will increase our dataset to include more food types other than the 14 types we experimented with here.

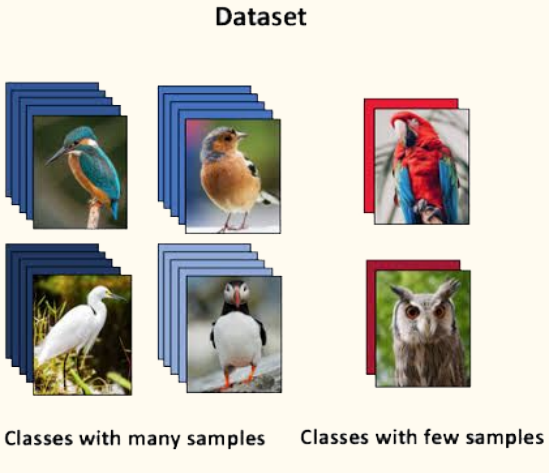
ii. We will extend our system to handle the more realistic scenario where the user provides an image of a meal rather than just one individual food item as we assumed here.



iii. We will remove the restriction of two input images from top view and side view. So that we can estimate food size with only one single using 3D/2D model-to-image registration.

iv. **Why should I trust you**

LIME (Locally Interpretable Model-Agnostic Explanations)an algorithm that can explain the prediction of *any* classifier or regressor in a faithful way by approximating it locally with an interpretable model. For image classification, an interpretable representation may be a binary vector indicating the "presence" of a super-pixel.



v. **Few Shot Learning**

In typical learning (on a single dataset), each individual sample-target pair functions as a training point. To ensure that an ML framework can exhibit similar behavior, we have to train it on multiple tasks by themselves— thus making each individual dataset a new training sample.

Conclusion

We adapted a pipelined approach that first predicts the type and size of the food item in the image, then uses this information to predict the amount of calories, fats, carbs, protein and sugar in the food item.

All our prediction tasks were performed using deep learning with some standard pre-trained model like InceptionV3 for object classification and YOLOv3 for object detection in fast food images.

We compared our pipelined approach to a baseline approach that directly predicts the amount of calories based only on the image, and showed a reduction in prediction accuracy.


Thank you & enjoy Fast Food!
Coke
NUGGETS
fried