Image-Based Machine Learning Nutrition Label Scanner

By Adhil Akbar, Adam Piaseczny, Rohan Joshi, Jingyuan Liu, Aarini Panzade, Jiarui Xie | Mentored By: Dr. Edward J. Delp, Carla Zoltowski

Abstract

The purpose of our project is to develop an android application that utilizes image-based machine learning to positively help impact an individual's health. A healthy diet is essential for a good healthy lifestyle and helps protect against many chronic noncommunicable diseases. The application we are creating, will allow users to set a health goal and take a photograph of a nutrition label. After doing so, the image will be uploaded to a Purdue based server, where an image-based machine learning algorithm will be used to extract the data from the label. This extraction will make use of image segmentation, cropping and a variety of pixel manipulation algorithms. The data extracted will consist of key nutritional information such as calories, fats, cholesterol, sodium, etc. This data will then be analyzed and measured using a health check algorithm to determine if the selected food is healthy for the user based on their goal. The Harris-Benedict formula and Mayo Clinic health guidelines will be incorporated within this algorithm. After determining this, the user will be notified whether the item they scanned is healthy for them. In a time when the population is plagued by diseases such as cancer, diabetes, and heart disease, we hope that this application can help individuals lead a healthier lifestyle.

Introduction

We have sought out to create an application that can help users make healthier choices regarding the types of food they eat. Upon opening our application users will be prompted to enter their health settings and complete a health profile. The health settings consist of Blood Pressure, Heart Health, Cholesterol, Weight Loss, Muscle Gain and Diabetes. The health profile consists of the user's height, weight, gender, and activity level. Next, the user can take a picture of a nutrition label. Once taken, the image is uploaded to our server, where our image-based machine learning algorithm is run to extract the label's data. Once extraction is complete, the data is sent back to the application, where our health check algorithm utilizes the user's health profile to solve the Harris-Benedict Formula. This formula yields the user's BMR. Using this value and the percent daily intake values, the health check algorithm will inform the user whether the item they scanned is a healthy choice for them.

Methodology

- The application was constructed using Android Studio and is coded in Java.
- Different Android phones were used to ensure universal compatibility of the application.
- A Purdue based server was used to upload scanned nutrition labels and run the machine learning algorithm.
- Volley and HTTPS was used to upload data to the server.
- Python and PHP was used to manipulate database directories.
- 200+ images were taken of various nutrition labels to create our training set database. Using this, we were able train our model. Images taken consisted specifically of FDA approved labels and did not consider foreign labels. The next step after completion of this project is to account for foreign labels.
- Machine Learning Algorithm was written using Python, Keras and TensorFlow.
- Health Check Algorithm was written using Java.

Significance of Research

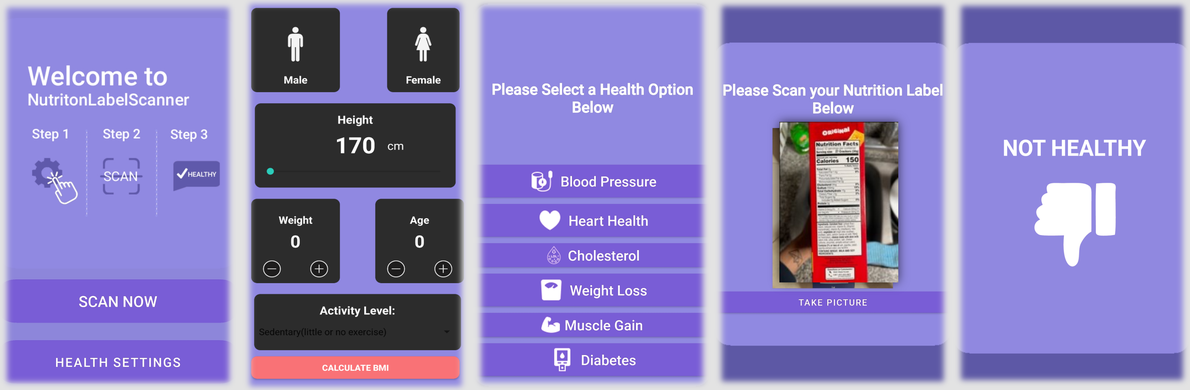
We are living in a time where the population is plagued by diseases such as cancer, diabetes, heart disease and many other health risks. Through our use of image-based machine learning and a customized health check algorithm, we intend to help users better understand the type of food they consume. We hope that through this application we can guide individuals to follow a healthier lifestyle.

References

[1] https://developer.android.com
[2] https://www.mayoclinic.org
[3] https://www.heart.org
[4] John Canny, A Computational Approach to Edge Detection (1986) https://skynet.ecn.purdue.edu/~ace/vip/A_Computational_Approach_to_Edge_Detection.pdf
[5] Sriram Baireddy, Daniel Montserrat, Edward Delp, Introduction to Deep Learning https://skynet.ecn.purdue.edu/~ace/deep-learning/tutorial-lecture
[6] Zaynab Kattabi, Younness Tabii, Abdelhamid Benkaddour, A New Morphology-based Method for Text Detection in Image and Video https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.800.4732&rep=rep1&type=pdf
[7] Yann LeCun, Yoshua Bengio, Convolution Networks for images, Speech, and Time-Series http://www.iro.umontreal.ca/~lisa/pointeurs/handbook-convo.pdf
[8] https://www.deeplearningbook.org

Application

Figure 1: Application Layout



Welcome Page | User Health Profile | Health Settings | Scan Label | Is Item Healthy?

Algorithm

Figure 2: Algorithm Steps

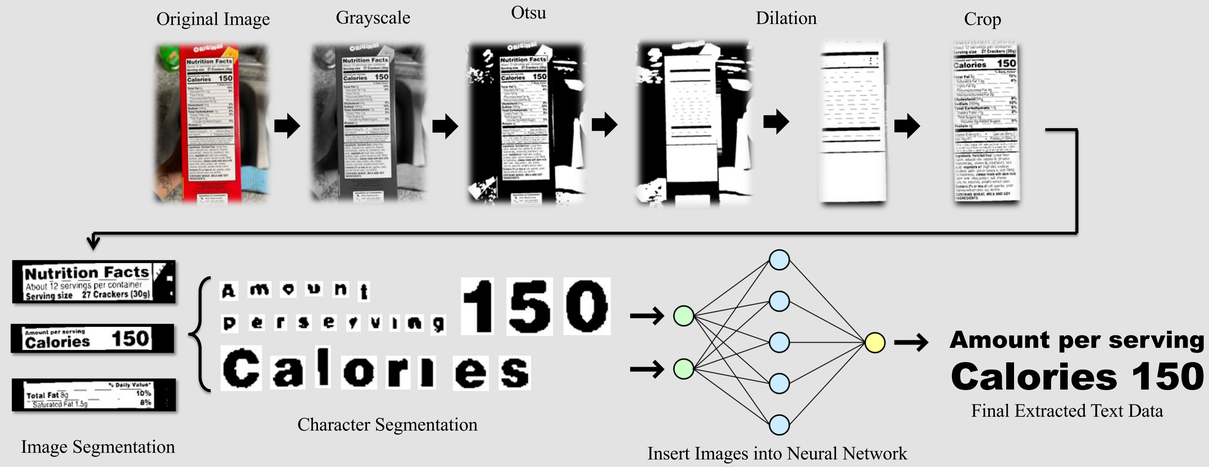


Image Segmentation | Character Segmentation | Insert Images into Neural Network

Harris-Benedict Formula

This formula is used to calculate an individual's Basal Metabolic Rate (BMR). BMR is the amount of energy (calories) your body needs at rest.

Formula:
Male = 66 + (6.3 x body weight in lbs.) + (12.9 x height in inches) - (6.8 x age)
Female = 655 + (4.3 x weight in lbs.) + (4.7 x height in inches) - (4.7 x age)

The resulting value is multiplied by based on activity level. Ex. (light exercise/sports 1-3 days/week) : Calorie Calculation = BMR x 1.375

Otsu's Method

Otsu's method is used to perform automatic image thresholding. This method returns a single intensity threshold that separate pixels into two classes consisting of foreground and background.

$$\sigma_B{}^2{}_k = P_1 P_2 (\mu_1 - \mu_2)^2$$

k: Threshold
P_n: Proportion of group n values less than k
μ_n: The mean value of group n pixels less than k

The higher between-class variance, the more different the two groups are. This allows us to separate the "foreground" and "background".

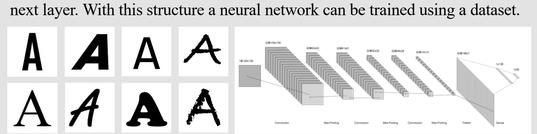
Image and Character Segmentation

Segmentation is achieved using a bounding rectangle and histogram projection.

This yielded a detection rate and recognition accuracy of 98%.

Neural Network

A neural network consists of multiple layers of nodes (neurons) connected to each other which each have an associated weight and threshold. If the input to a node exceeds the threshold the node will activate and send the data to the next layer. With this structure a neural network can be trained using a dataset.



The network was trained using a dataset of A-Z, a-z, and 0-9. The dataset contains a total of 63,055 images of size 128 x 128 with multiple different fonts for each character.