Wilbert Guo
1002990720
wilbert.guo@mail.utoronto.ca

CSC420 Final Project Report
Developing a Nutrition Label Extractor

1. Introduction

In today's society, people are becoming more conscious of the decisions they make in their day to day life. One of these such decisions is picking the right food to eat based on the nutrition labels presented on each food product and tracking how many calories are consumed per meal. With so many food options now, consumers sometimes find it difficult to choose one brand over another when buying specific food products.

1.1. Problem Overview

Chances are, everyone has tracked their diet and nutrition at some point in their life and realized how tedious the manual input of data can be. Additionally, some consumers would like a more visual representation of the nutrition info presented on these labels so that they can compare the nutrition categories of a specific food product to see which one is best suited for their diet. For example, when shopping for a breakfast cereal, the customer might want to know which brand of cereal has the largest ratio of protein to sugar. Traditionally, this would have to be done manually by inputting the nutrition label information into a data visualization software plotting charts. This approach is not a sustainable approach and certainly not optimized for nutrition label information extracting.

1.2 Proposed Solution and Hypothesis

The solution that we proposed is to use a combination of computer vision techniques as well as leveraging cameras built into smart phones that most people have in today's society. The aim is to allow consumers to use their smart phones to take a picture of the nutrition label and have an output of a pie-chart showing the breakdown of the major nutrition categories. We expect the text recognition step (OCR) to not be 100% accurate some of the time since we plan on training a CRNN and that requires tweaking parameters to best suit the situation.

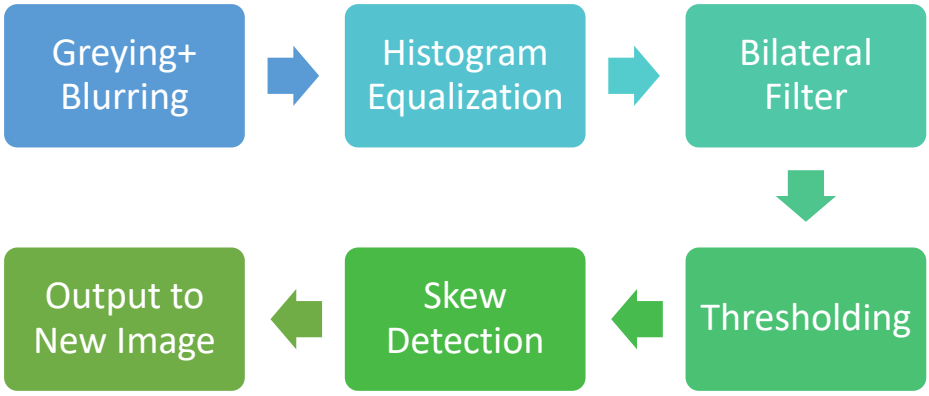
1.3 Related Works

We got our inspiration for this project from this research paper: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=8204&context=etd.

In the paper the process of steps includes first taking a picture of the nutrition label using a smart phone. Then the captured image is processed through a number of steps include separating out the label from the background and rotating the image if required. Once the image is preprocessed, it is then fed to a CNN for text recognition where the alphabetical letters and numerical values are extracted from the label.

2. Methods

2.1 Pre-processing the Image for OCR (Wilbert)



The pre-processing stage is the first step where we pass the raw image of a nutrition label taken by a consumer and we perform a series of image transformations. The first image transformation is converting the image to greyscale and applying a blur. Converting to greyscale allows the image pixel data to have only one value ranging from white to black while applying a blur will help smooth out any outlying pixel values. The next step is to do histogram equalization which helps increase the contrast in areas that have different intensities. This step essentially darkens the nutrition label border and creates a more noticeable distinction between label border and the lighter intensity background. After applying histogram equalization, we apply a bilateral filter to the image to create more uniform pixel areas in the image background. This filter helps the background area around the label border evens out to a lower intensity than the border itself. Next we do image thresholding. The purpose of this step is to set the very dark pixels to black and the whiter pixels to white. Essentially this step returns a pure black and white image. After this step is complete, we finally apply skew detection to the image in case the words are at an angle. Once we have finished all these steps, we output the new image and pass it along to the OCR step. Originally, we also wanted to implement contour detection and image segmentation to help extract only the label part from the image however, we realized that it would include training a neural network to recognize the label and we ran out of time.
The code for this step can be found in the file: CSC420 Final Project PreProcessing.ipynb

2.2 Optical Character Recognition (David)

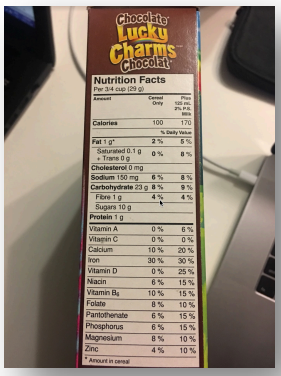
The original plan for this step was to use a CRNN which takes as input the pre-processed image and outputs a string with the data from the label. We realized that our CRNN neural network does not work as well as we had planned so we explored other options and, in the end, decided to use pytesseract. This is based on the OCR engine Tesseract. This takes in a pre-process image as input and also outputs a string which takes into account text being on different lines. Taking account of text on a different line was something that our CRNN could not do.
The code for this step can be found at: CSC420 Final Project Post-Process.ipynb

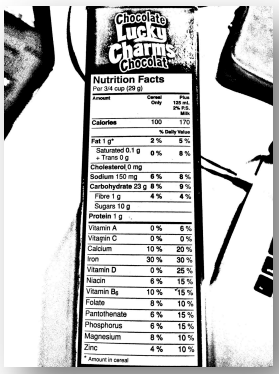
2.3 Post-Processing Step (David)

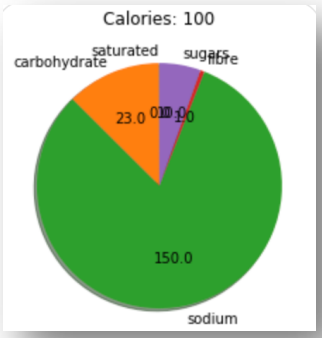
In this step, we take the output from pytesseract and clean up the raw OCR text and then match it with possible desired words that we have included in na_nutritients.txt. This list accounts for the nutrient categories of both USDA food labels, Canadian food labels and French keywords. We then use a bi-partite graph matching algorithm where one set of the graph vertices is the desired words while the other set of vertices is the possible candidate words after cleaning up the OCR output. The edges the graph represent the Levenshtein distance between the candidate word and the desired word. The goal of the algorithm is to find the minimum distance between all the candidate words. The final step in the post-processing stage is to output the information into a pie chart and also display the calories.
The code for this step can be found at: CSC420 Final Project Post-Process.ipynb

3. Results and Discussions

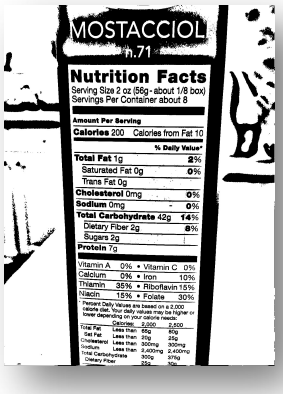
For our results we got some pretty accurate results from some nutrition labels and less accurate results for other nutrition label. In the nutrition label below we can see that this one is fairly accurate since it contains most of the nutritional food groups but is missing protein. One other challenge that we did not foresee is the cluttered annotations on the pie chart. Even though the starting image was slightly blurry, I think the pre-processing steps of the image before feeding in to the OCR engine was a huge help in identifying the alphabetical characters and numerical values from the nutrition label.







The images below are another test label that we tested our program one. This result seemed to be less accurate than the previous result since the dietary fiber is off. However, the total calories and the carbohydrates seem to be accurate in the pie chart. Another nutrition category that was not extracted was protein and I think it is due to the lighting being poor and having less contrast between the text and the white background.



4. Main Challenges

4.1 Bad Input Pictures

One of the main challenges is dealing with input images that are either blury or have bad lighting. When the input image was as such, the output from the OCR engine was not accurate and had a hard time recognizing the text in the nutrition label. However, we did realize that pre-processing the image did indeed help a lot with getting more accurate readings of the text and numerical values from the nutrition label.

4.2 Wasting Time on Research

Another one of the challenges we faced was spending too much time trying to get the CRNN working for the optical character recognition step. Originally the CRNN had only a few letters as output and it was clear that there was a lot more text present in the nutrition label. In addition, we also spent a lot of time trying to debug the methods that were present in the paper and I feel like this is not the best approach to implementing new things. I think that we should have spent more time trying to implement the solutions presented in the paper and play around rather than fully trying to understand what the research paper was explaining.

5. Conclusion

In conclusion, I think that our project was not entirely finished since there was still aspects of it that could be improved such as including contour detection, image segmentation and training our own neural network for optical character recognition. Just to summarize again, the workflow model that we currently have is taking in a raw input image taken by the consumer and performing image processing techniques to get a new image where the text and numerical values are more easily identifiable. Then we pass this new image to our OCR engine and extract the text and numbers were we then output to a pie chart using python. I believe this project is a good starting point as a potential prototype to other projects that would like to extract the nutrition information from nutrition labels.

6. Citations

(https://www.researchgate.net/publication/283727396_Image_enhancement_by_Histogram_equalization)

(https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Tomasi98.pdf)

(https://pdfs.semanticscholar.org/3796/3f1672c4f53f57a1774c381aaf86695f2087.pdf)

(https://www.researchgate.net/publication/220494200_Adaptive_Thresholding_using_the_Integral_Image)

(https://www.pyimagesearch.com/2017/02/20/text-skew-correction-opencv-python/)

(https://pypi.org/project/pytesseract/)

(https://arxiv.org/pdf/1507.05717.pdf)