EXTRACTION OF NUTRITIONAL INFORMATION USING IMAGE PROCESSING

KHUSHAL PUJARA (RIT STUDENT)

ABSTRACT. In today's world, daily nutrition tracking requires constant watch on numerous nutritional metrics that includes both macro and micronutrients. It requires undesirable amounts of either time or man- power. Available options include tediously recorded and updated nutritional information and unreliable crowed sourced or costly maintained database. This project will overcome these obstacles by providing an interface which will use computer vision and image processing techniques and will read and extract nutritional data information from the nutritional label itself.

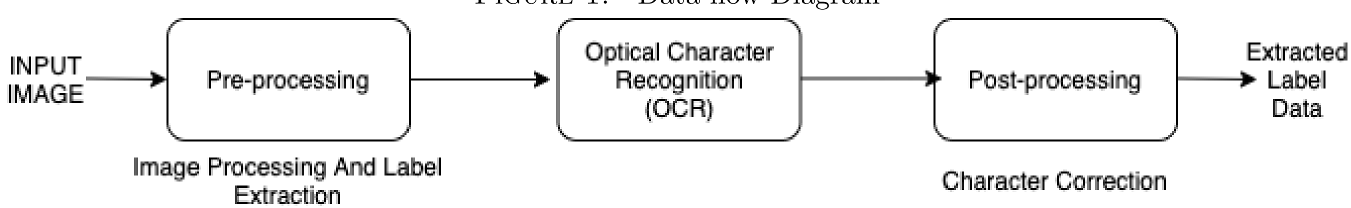
1. OVERVIEW

The final goal of the program is to detect the nutritional label and extract the text present on those label. We will be specifically looking for things like:

- Total Carbohydrate
- Total Fat
- Saturated Fat
- Dietary Fiber
- Sugars
- Protein

In order to achieve goal, we will be following the below data flow model.

FIGURE 1. Data-flow Diagram

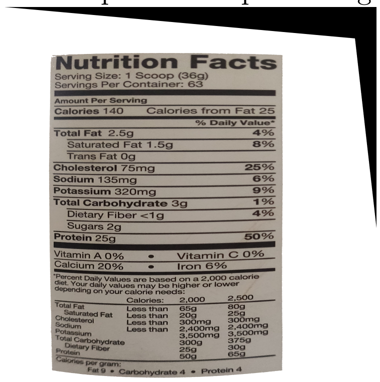


1.1. Pre-processing.

The input image first goes through the pre-processing module.In this part our goal is to identify and separate the label from the rest of the image. Different types of contour detection could be used to do that. My goal was to use just the morphological operations and complete the task. This step is responsible for Image cleaning, filtering, boundaries detection, shape detection etc. The figure below shows us the output of the first part.

Date: 04 August 2020.

Figure 2. Output of Pre-processing module



1.2. **Optical Character Recognition.**

The task is to convert the label image obtained from the pre-processing unit into binary image and reduce the complexity to apply OCR on the image. OCR then attempts to identify the actual English words and number values and shows it as the output. There are many commercial OCR engines available in the market. For project purposes we are using Matlab's OCR command.

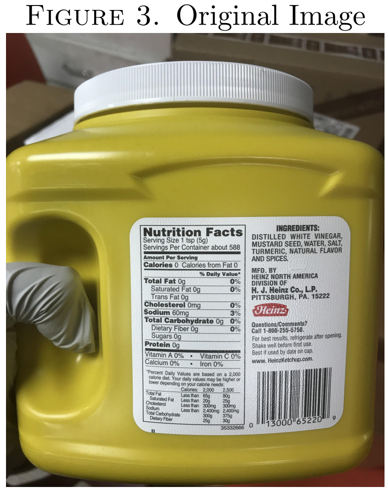
1.3. **Post-processing.**

This module corrects the errors and inaccuracies of the OCR output. It formats the data and displays the data into a readable form. As we are not using any commercial OCR modules, Matlab's free module might give errors such as misspellings or misidentified characters that might not be on the original label. This unit analyzes the texts and collects important information while ignoring the unimportant text.

2. Conceptual Design

This section will give a detailed explanation of the techniques we used in our pre-processing and post-processing module

Figure 3. Original Image

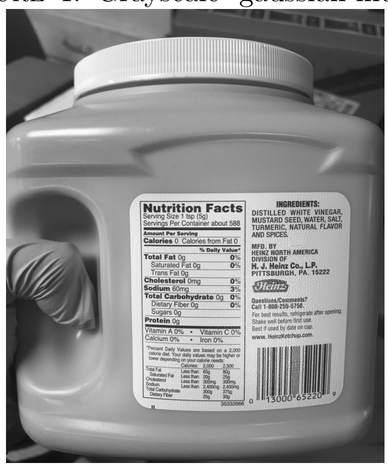


2.1. **Pre-Processing Stage.** As mentioned above in section 1.1 the objective of pre-processing stage is to identify and separate the label from the rest of the image To help accomplish this goal, i used some morphological and filter operations from Matlab. This chain could be broken in 6 distinct stages (listed below) that run sequentially. Each stage is accompanied by an example of a sample image processed with our program.

2.1.1. *Noise removal and blurring the image.*

- The first step is to convert the image from a multi- color space to grayscale colorspace. Most of the colored images are represented using three values per pixel, a red hue value, a blue hue value, and a green hue value. Converting to grayscale transforms the pixel data to only depend on a single value ranging from white to black with shades of gray in between. Once the image has been grayed, we apply a subtle blur to help smooth out any outlying pixel values. The blur looks at each pixel and alters its value so its intensity more closely matches the relative intensity of the neighborhood of pixels around it.

Figure 4. Grayscale- gaussian filtered



2.1.2. *Contrasting and Filtering.*

- The next step is to increase the contrast of the image to separate the nutritional label output from the background. As distinct the border will be, easier will be to detect the contour. For increasing the contrast we did adaptive histogram equalization. As there is still some noise in the image, we applied the filter again. But at the same time, preserving the edges of the nutritional label was really important. Bilateral filter was a perfect fit for these kind of scenarios. It removes the noise at the same time it preserved the edges.

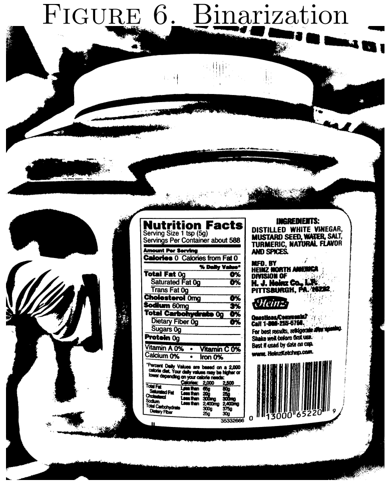
Figure 5. Adaptive Histogram - Bilateral Filter

2.1.3. *Binarization of the image.*

- Previous steps performed all the necessary morphological operations to visually separate out the label from its background. The next step is to convert the gray-scaled image into binary image.

Binarization of an image converts the image into 0-1 pixel values using a certain threshold. As out label is white and the boundaries are black anyway, we don't need to give a threshold for the binarization process.

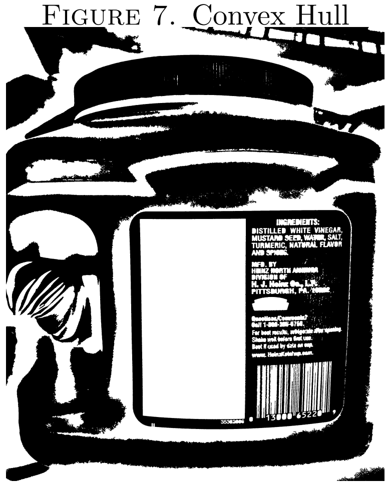
Figure 6. Binarization



2.1.4. *Reducing the search space.*

- The label we are looking for will occupy approximately 30 - 40% of the image area. For reducing the search space we can fill in the area which occupies less than 20 percent of the total area of the image. That can be done using bwareaopen command. bwfilter will also help to do the same thing. Applying convex hull on the pre processed image will return the following output.

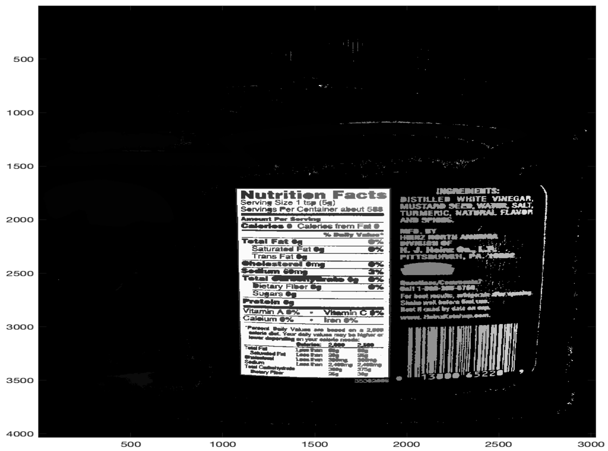
Figure 7. Convex Hull



2.1.5. *Capturing the boundaries.*

- We found out the strong boundaries after getting a number of convex hulls in the image. As we have already reduced the search space, we will find all the enclosing space (boundaries) in the image using bwboundaries. This function will give all the enclosing pixels of an enclosing boundary same value. As you can see the figure, the pixels in the nutrition label will have the same value

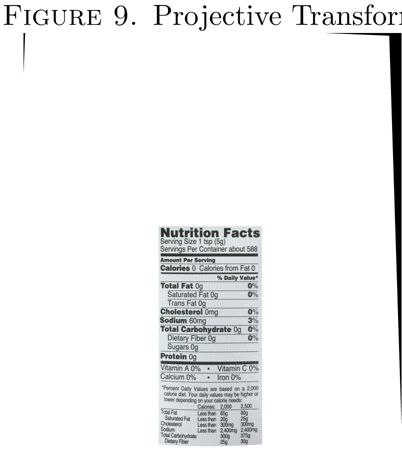
Figure 8. bwboundaries



2.1.6. *Selecting the right boundary & applying projective transform.*

- After getting all the boundaries, we need to select a boundary which contains the label we need. In an average image we can find almost 1000 boundaries. As I previously mentioned, the label will occupy around 30 - 40% of the image area. I sorted the boundaries with respect to the area. The boundary we care about must be in top 5 boundaries. We apply a shape recognition algorithm on top 5 boundaries we decided to keep. The boundary which comes out to be rectangle is our label.
- After getting the correct boundary, we apply the corner detection algorithm to get the four corners of the label.
- The last step is to apply projective transform on the boundary corners and get the label with minimal distortion.

Figure 9. Projective Transform



2.2. **Post-Processing Stage.**

As mentioned above in section 1.3, the purpose of the post-processing stage is to find and correct any errors in Matlab's OCR output. As we are analyzing the food labels, we can expect a restricted list of characters and words to appear on the label. Using this domain knowledge we can increase the effectiveness of the whole process and accurateness of the final result.

2.2.1. *Cleaning the OCR output.*

- As we are not using commercial OCR software, the OCR output we get could be inaccurate. Some of the characters could be misread. Some of the words could be misspelled. Some of the spaces could be missed. We will try to clean the output as much as possible

2.2.2. *Key-word matching.*

- We already have a list of possible words that could be present in the found nutritional label. We can traverse through the text output given by Matlab's OCR function and see if words are matching with our set of possible words.

3. Results

- As the label extraction is completely based on image processing, we found an accuracy rate of 90%
- Though label extraction task was very successful, i faced some issues with OCR function of Matlab. Matlab OCR function was successful in detecting all the characters and number. But unlike some commercial OCR softwares, Matlab OCR was not so successful in mapping the nutrition category with it's number.
- After using certain hardcoding of categories, i was able to get some success rate, but it was not upto mark.

4. Discussion and Lessons Learned

This was probably the most interesting project i have ever worked on. In the beginning, when the course had just started, it seemed kinda easy - I felt like how hard could it be to extract what we can see visually. As the course continued i gradually learned different morphological operations which i tried applying on the nutrition label images.

Gradually i started understanding the complexities and issues. But i was used to finding different things from Matlab forums and stack overflow. In between i also tried to learn and apply a neural network which was not so successful. I somehow continued with only image processing. Topics like sorting, bwboundries, bilateral filtering were unfamiliar for me. With a specific task, with internet suggestions i learned using them.

I was done with label extraction part 15 days before deadline and felt like OCR would be really easy. I underestimated the task and its complexity and now i couldn't get nutritional category mapped accurately with its quantity. It could've been done if i use other commercial OCR functions supported by Matlab. But it was too late already.

5. Future Work

Post processing chain of the program could be improved dramatically. The current mechanism of mapping of category and its value did not turn out to be so successful. If we could somehow manage to do OCR vertically and horizontally as well, it would be helpful to get the correct nutrition category mapped with the correct value. With some commercial OCR functions that could be easily done.

6. Conclusion

The goal to apply computer vision techniques in the field of daily nutritional tracking was achieved successfully. Using domain knowledge and MATLAB's basic morphological and image processing techniques we built an innovative program that could be proven better than existing cumbersome nutritional tracking models. As i did not have any prior knowledge of Computer Vision or MATLAB, project was a bit heavy. With trial and error, i integrated different types of image processing algorithms and made the label extraction work with 90% of accuracy. Though the computational complexity can still be reduced with applying certain rules. I want to thank Prof. Thomas Kinsman for giving me an opportunity to work on this very interesting project. As i mentioned in the discussions, with using correct OCR algorithms, this project could run very efficiently in the market.