M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Automation of a fruit sorting system

Andri Geir Arnarson (s164518)
Asger Frederik Græsholt (s154099)

Supervisor: Nicki Skafte Detlefsen
Kongens Lyngby 2022

Abstract

Decleration

Andri Geir Arnarson (s164518) Asger Frederik Græsholt (s154099)

Kongens Lyngby, March 29, 2022

Contents

CHAPTER 1

Introduction

1.1 Motivation and backstory

Food waste is one of the bigger challenges modern society faces nowadays. In a study from 2013 made on the footprint of food wastage [3] by the Food and Agriculture Organization (FAO) it is estimated that approximately 33% of all food produced for human consumption is wasted or lost along its way to the costumer. This is not only tragic in the light of famine across the world, but this also results in excess amounts of CO2 emissions as well as huge losses of revenues for many companies. Food waste can be found in all layers of the food supply chain but is mostly visible in the trash bins of the producers, distributors, and supermarkets. Package failure, excess ordering amounts, or single items gone bad in packages of several are all examples of scenarios where the product, or at least a big part of the product, is in perfect shape but are thrown out because of the extra work needed to preserve or restore it's value. The aim of this project is to automate a big part of this extra work needed to reduce cost barriers for upcycling food. By an implementation of a fully automated fruit sorting system this study will work as a prove of concept (POC) to demonstrate the potential of applications that can support a future with less food wastage. This also complies with several of the United Nations the Sustainable Development Goals (SDGs)[REFERANCE].

1.2 Collaborators

The projects main collaboration partner is the Icelandic upcycling start-up Humble ehf[2]. Humble works closely with fruit and vegetable distributors as well as supermarkets in Iceland where they try to exploit the hidden opportunities that lie within food-waste solutions and upcycling of food. Currently Humble is collecting fruits that normally would end up in the trash bin. Each fruit collected is instead inspected and if deemed good by quality control used in a new product. An example could be boxes of strawberries that are thrown out because a few berries have gone bad. Humble takes these berries that are still in good condition and freezes them. Those frozen strawberries can then be packaged and sold back into supermarkets.

LEGO mention needed

1.3 Objectives

This process of sorting bad fruits from good ones is very labour heavy when done manually but is very well suited for automation. Similar tasks are currently performed in the food industry but mainly focusing on fresh ingredients. The task at hand is to enable automatic classification and discard of rotten fruits. One of the most obvious differences between rotten and fresh fruits is the visual difference which is why the project take its offset within the domain of image recognition. To simulate a real-life scenario a prototype of an industrial system will be made with a conveyor belt to pass the fruit through the system. In order to realize a functional prototype of such a fruit sorting system, quite a few components have to be developed and implemented in sync. The project is therefore split up into three major phases: Data generation, Image recognition modelling, and Physical prototype development. The rest of the report will also follow this structure.

CHAPTER 2

Related work

CHAPTER 3

Phase 1 - Data generation

The first phase of the project evolves around collecting real life data. The images of the Kaggle [5] dataset described in section 4.1 are plentiful but are not completely comparable what a camera would see on a production line in real life. A data collection/generation system was therefore setup in collaboration with Humble in order to collect images of both rotten and fresh fruits in a production line environment.

3.1 Physical setup of the data collection system

The collection systems main focus was to generate quality images that would reflect the actual production line, but it also had to be efficient and easy to use for persons with little technological background. First in order to ensure that photo quality was good a smart phone was the perfect fit. Today's smartphones are equipped with very good cameras at a very affordable price, additionally smart phones are equipped with bluetooth, wi-fi and other general components suited to transfer the images from one device to another. In order to ensure constant lighting the smartphone was placed in a small lighting stand. The setup is shown in figure 3.1

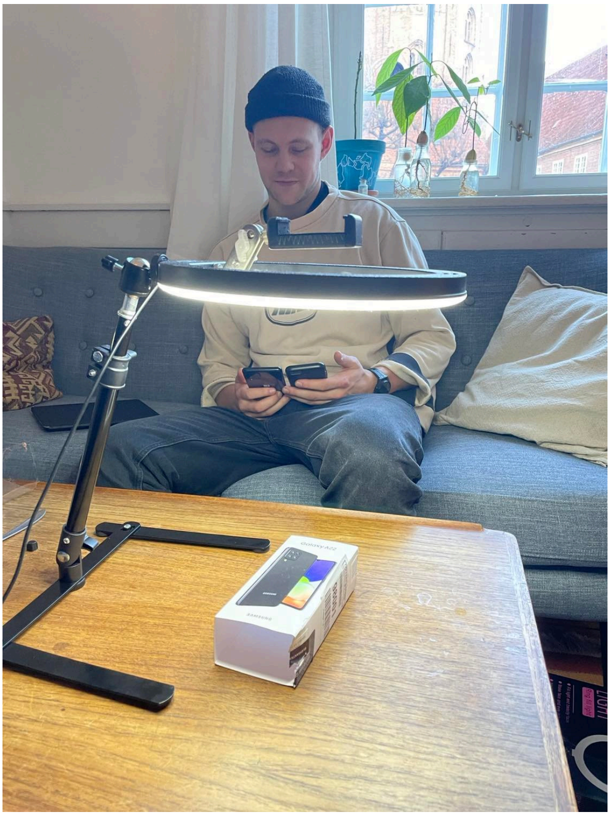

Figure 3.1: The data collection system.

3.2 Mobile Application

The physical setup provides all the tools needed to take the image but the physical setup is located in Iceland and an easy remote access to the images is needed as well. The smart phone is equipped with both wifi and touch screen that allows for easy user interaction, therefore a mobile application served our purpose perfectly. The mobile application was a simple one but simplified the process significantly. The wireframe flow can be seen in figure **??**.

First the user is prompted with a screen asking what kind of fruit is being processed today, then the camera screen comes up until the user takes a photo and is prompted with a new dialog asking whether its rotten or fresh. Finally the image is stored in a Firebase [4] database while organized by both fruit and freshness.

WIREFRAME FIGURE

3.3 Quality tests and results

We have not written anything since we are still in the generation phase and hence do not have sufficient amounts of data to test upon.

CHAPTER 4

Phase 2 - Modelling

Analysing every pixel in an image taken by a 48MP camera[1] (as used in the setup explained in section 3.1) is an unrealistic task for a real time system, thus alternative approaches must be considered. In this phase the core functionality of the system is explained in detail along with core elements related to detection and classification of the fruits passing through the machinery.

4.1 Data

The data used to train the initial versions of the classification models originates from Kaggle [5] and is publicly available. The data consist of images of apples, bananas, and oranges with both fresh and rotten examples for each type of fruit resulting in 6 different categories. The dataset is unlabelled but stored in a folder structure that preserves the distinction between classes. All images are pictures of fruit taken in controlled settings with single or multiple objects of the same type in the frame. The images range in size and resolution and are cropped to a certain degree to fit the captured object. To increase the size of the dataset augmentation (mirroring, rotation and cropping) has been applied, resulting in a total size of 13,599 images to train and test upon (see table 4.2 for detailed information).

4.2 Pre-processing

To feed the model with the data, certain pre-processing steps are required to generalize the data input. A csv file is generated and used for loading data into the model to link an ID/label to each image based on its file path and hence its category. The images are cropped to squares using the shortest dimension and the resolution is lowered to 128×128 (ref to justification?). Resulting in a constant input size of $3 \times 128 \times 128$ (channels, height, width). In the end linear normalization is applied to limit the input values to be within the interval $[0 ; 1]$.

[1] A camera with 48 Mega Pixel resolution can store images with up to 48.000.000 pixels

Categories	ID	Train	Test	Total	Pct. (%)
Fresh apples	0	1,693	395	2,088	15.35
Rotten apples	1	2,342	601	2,943	21.64
Fresh bananas	2	1,581	381	1,962	14.43
Rotten bananas	3	2,224	530	2,754	20.25
Fresh oranges	4	1,466	388	1,854	13.63
Rotten oranges	5	1,595	403	1,998	14.69
Total	-	10,901	2,690	13,599	100

Table 4.1: Basic stats of the fresh and rotten fruit dataset [5].

4.3 Convolutional Neural Networks

Convolutional neural networks(CNN) are types of neural networks that are specifically well suited for image recognition. CNN's assume that the input is an image which allows for a sensible structure suited for image recognition than other neural networks[6]. One might say that CNN's are built specifically for image recognition because it exploits the structure of an image, rather than inefficiently considering every single pixel. This section will go through the building blocks of CNN's but does assume general knowledge of concepts including but not limited to artificial neural networks, back-propagation, activation functions and a loss-function.

4.3.1 Convolutional Layer

The input of a CNN is assumed to be an image and is presented in a 3-dimensional arrays in the shape $[PIXELS \times 3]$ corresponding to the three RGB values of each pixel. Each value ranges from 0 to 255 depending on the brightness of that color channel.
Every convolution consists of a set of learnable filters, each with the aim of be able to detect a certain feature. The filters,shown in the middle of figure 4.1, are usually spatially small but cover the full depth of the input i.e $[F \times F \times D]$ weights, where F is the filter size and the D input dimension. These weights, along with a single bias, are the ones that are optimized to detect different features. To convolute means to roll upon it self and that is exactly what the convolutional layer does during the forwards pass where the filter slides across the inputs height and width. The dot product of the input and the filter is called an activation map, shown to the far right in figure 4.1. The activation map is a 2-dimensional and stores the response of that filter at each spatial position i.e. the activation map stores information weather and where a certain feature was detected by that filter.

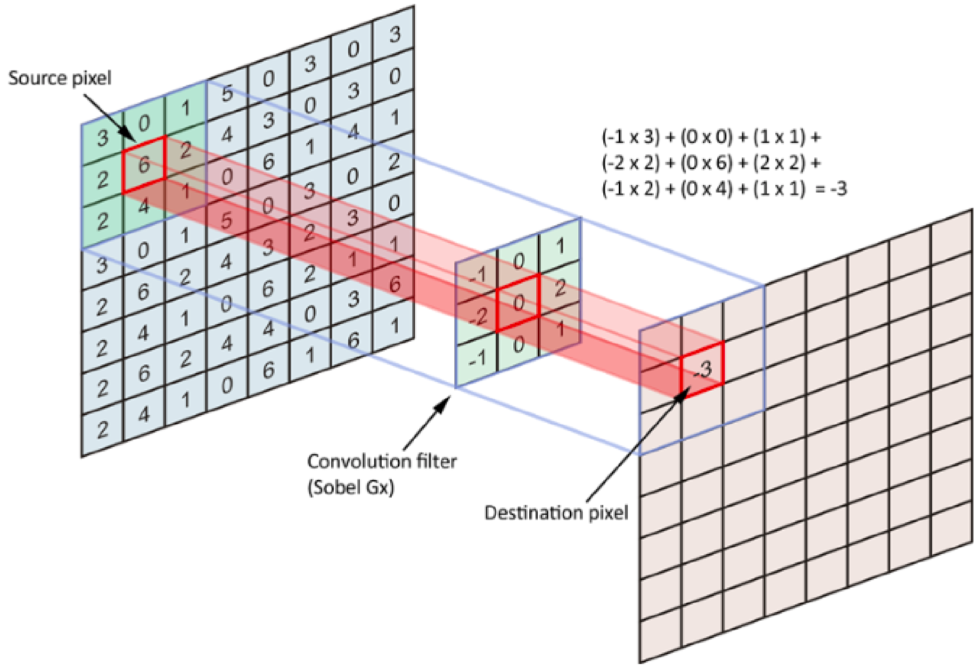

Figure 4.1: Convolution [1].

Finally each neuron on the convolutional layer is equipped with a non-linear activation function. This non-linearity is a very important feature as it allows the network to learn diverse and complex features and mappings. The most popular ones used in convolutional networks are the Sigmoid and ReLU functions0

4.3.2 Pooling Layer

4.3.3 Fully connected layer

4.4 Object Detection

4.4.1 Biggest Component

4.4.2 YOLO

4.5 Model testing and results

CHAPTER 5

Phase 3 - Implementation of physical prototype

5.1 Main objectives

Concrete , measurable goals from that start. Measuremenst can include mdoel accuracy, prototype overal speed, compactness of prototype, or performance of each induvidual component

5.1.1 Prototyping theory / Development

Here we plan to list the mythology of how the prototyping phase will be conducted in order to accomplish the main objective. Agile and UX methods described etc

5.1.2 Design of Prototype

Here we will show how one solution could look, this is also our starting point but might(and probably will) change during the iterations below.

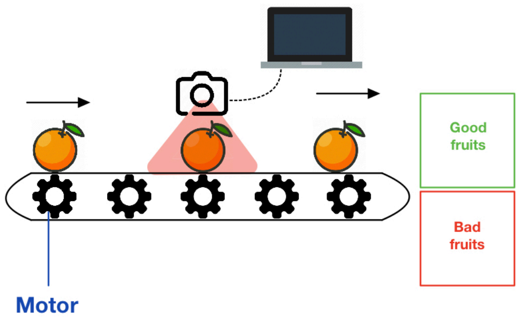

Figure 5.1: Prototype sketch.

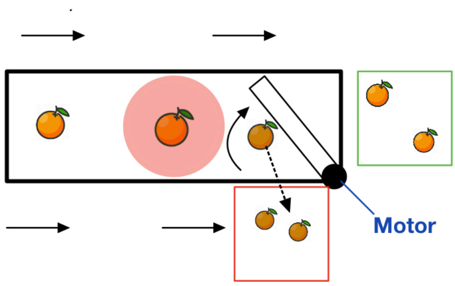

Figure 5.2: Prototype sketch.

5.2 Iterations

Each iteration will be structured around "Build-Measure-Learn" cycles from UX theory

5.2.1 Iteration 1

Build

Measure

Learn/Results

5.2.2 Iteration 2

. . .

5.2.3 Final Prototype(Last Iteration)

5.3 Pipeline flow explained?

5.4 Results and discussion

CHAPTER 6

Discussion

CHAPTER 7

Conclusion

Bibliography

[1] Shuchen Du. *Convolutional Neural Networks graph.* Accessed: 2022-03-23, Created:2020-01-02.

[2] Humble ehf. *Humble website.* `https://www.humble.is`. Accessed: 2022-03-17, Date created: 2022-01-15,

[3] FAO. *Food wastage footprint.* 2013.

[4] Google. *Firebase storage.* Accessed: 2022-03-22,

[5] Sriram Reddy Kalluri. *Fruits fresh and rotten for classification.* `https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification`. Accessed: 2022-03-17, Date created: 2018-08-24, Last updated: 2018-08-24, Current version Version 1, Visibility Public.

[6] Stanford University. *Convolutional Neural Networks (CNNs / ConvNets).* Accessed: 2022-03-23,