

A deep learning segmentation approach to calories and weight estimation of food images

Runar Isaksen
Eirik Bø Knudsen
Aline Iyagizeneza Walde

SUPERVISORS

Morten Goodwin
Vimala Nunavath

Master's Thesis
University of Agder, 2019
Faculty of Engineering and Science
Department of ICT

UiA
University of Agder
Master's thesis

Faculty of Engineering and Science
Department of ICT
© 2019 Runar Isaksen
Eirik Bø Knudsen
Aline Iyagizeneza Walde. All rights reserved

Abstract

Today's generation is very aware of what they are eating and the amount of calories in their food. Eating too many calories can lead to increased weight, which has become a big health issue. A study from 2016 states that more than 1,9 billion adults are overweight where almost one third of these are obese. Statistics from Norway show that 1 of 4 men and 1 of 5 women are obese.

Artificial Intelligence in general and deep learning in particular can be used to help understand the content of eaten food. In this master thesis, we propose a network to estimate the weight of food from a single image. This is done in three main parts: (1) image classification to classify what kind of food it is, (2) segmentation to segment out the different food from the image and (3) estimate weight of the food. This is then compared against a food database to get the calories. Both for the classification and the weight estimation is an inception network used, while a YOLO network is used for the segmentation. The solution is the first example of a working estimation of grams from a single image. The results for weight estimation give a standard error of 8.95 for all categories and 2.40 for bread which is the best category.

Keywords : Calorie estimation, Deep learning, Food classification, Image classification, Inception networks, Segmentation

Preface

Calories and weight estimation for food images using machine learning is a result of a master thesis project developed by Aline Iyagizeneza Walde, Eirik Bø Knudsen and Runar Isaksen. This work is done in the connection with master's thesis in IKT 590 spring 2019. The aim of the project is to train a machine to identify type of the food item in the image and afterwards predict the amount of calories in the given food item. We wish to take this opportunity to thank Associate Professor Morten Goodwin for very good guidance throughout the duration of the project. We are also very grateful to Postdoctoral fellow Vimala Nunavath which has been providing us support during the project period.

Grimstad

23.05.2019

Aline Iyagizeneza Walde, Eirik Bø Knudsen and Runar Isaksen.

Table of Contents

Abstract	iii
Preface	iv
Glossary	x
List of Figures	xii
List of Tables	xiii
1 Introduction	1
1.1 Motivation and Problem Statement	2
1.2 Research Questions	3
1.3 Thesis Goals	4
1.4 Assumptions and Limitations	4
1.4.1 Assumptions	4
1.4.2 Limitations	5
1.5 Contributions	5
1.6 Thesis Outline	6
2 Background	7
2.1 Artificial Neural Networks	8
2.1.1 Perceptron	8
2.1.2 Activation Functions	9
2.1.3 Optimization	10
2.1.4 Deep Neural Network	10
2.2 Convolutional Neural Networks	11
2.2.1 Convolution	11
2.2.2 Pooling	12
2.2.3 Flattening	13
2.3 Residual Neural Network	14

2.3.1	Residual Learning	14
2.3.2	Residual Block	15
2.4	Image Segmentation	16
2.4.1	Region-based Segmentation	16
2.4.2	Edge-based Segmentation	18
2.4.3	Segmentation with intersection over union	19
2.5	You Only Look Once	19
2.6	Diet	20
2.6.1	Nutrition	21
2.6.2	Calories	22
2.6.3	Food Table	22
3	State of the Art	25
3.1	Food Recognition and Classification	26
3.2	Food Segmentation	31
3.3	Weight and Calorie Estimation	34
4	Methodology	37
4.1	Dataset	38
4.1.1	Food-11	38
4.1.2	Food-101	38
4.1.3	FoodX	38
4.2	Data Pre-processing	39
4.2.1	Prepare Food-101	39
4.2.2	Prepare FoodX	40
4.2.3	Bounding Box	40
5	Experiments and Results	43
5.1	Network Architecture	44
5.1.1	Food Recognition and Classification	44
5.1.2	Segmentation	45
5.1.3	Weight and Calorie Estimation	47
5.2	Results and Discussions	48
5.2.1	Food Recognition and Classification	48
5.2.2	Segmentation	51
5.2.3	Intersection over Union	52
5.2.4	Weight and Calorie Estimation	55
6	Conclusion and Future Work	61
6.1	Conclusion	61

6.2	Future Work	62
6.2.1	Increase Dataset	62
6.2.2	Using Food Table API	63
6.2.3	Use of Included Ruler	63
	References	70
	Appendices	71
A	Code repository	72
B	Weight results	72
B.1	Results from Validation Set	72
B.2	Results from Training Set	76

Glossary

AI Artificial Intelligence. 1, 48

ANN Artificial Neural Network. 8, 10, 11, 13, 14, 29

BoF Bag-of-Feature. 27

ConvNet Convolutional Neural Network. xi, 4, 11, 12, 14, 15, 27–29, 32, 34

DCNN Deep Convolutional Neural Network. 31

DNN Deep Neural Network. xi, 3, 10, 29

FV Fisher Vector. 27

GNB Gaussian Naive Bayes. 29

GPU Graphical Processing Unit. 5

HoG Histogram of Oriented Gradients. 26, 27

IFV Improved Fisher Vector. 27

IoU Intersection over Union. xiii, 19, 32, 46, 53–55, 62

JSON JavaScript Object Notation. 46

MKL Multiple Kernel Learning. 26

ML Machine Learning. 3, 5, 8, 44, 48, 62

R-CNN Regional Convolutional Neural Network. 29, 33, 54

ReLU Rectified Linear Unit. 9, 15

ResNet Residual Neural Network. xi, 1, 3, 14, 28, 29, 45, 48, 50, 61

RF Random Forest. 29

SVM Support Vector Machine. 26, 27, 29, 34

XML Extensible Markup Language. 40, 41, 46

YOLO You Only Look Once. iii, xi, 1, 3, 19, 20, 37, 45, 51, 54, 62

List of Figures

2.1	A single perceptron [6].	9
2.2	A DNN with two hidden layers [9].	10
2.3	Overview over the ConvNet [10].	11
2.4	Converting an image into a feature map [12].	12
2.5	Add max pooling to the feature map [13].	13
2.6	A 34-layer deep ResNet [14].	14
2.7	Residual block [14].	15
2.8	Show region growing [18].	17
2.9	Example of region splitting in an image [18].	17
2.10	Different types of edges [20].	18
2.11	Illustrate the difference between segmentation methods.	19
2.12	Show how YOLO works [21]	20
2.13	How a varied diet could look like [24].	21
4.1	Example of how an image in the dataset look like, with 38g bread, 44g butter, 64g cheese, 52g melon, 40g mango, 45g grapes.	39
4.2	Show how weight is presented in the images.	41
4.3	Hiding a post-it note from the image.	41
5.1	Pipeline showing the proposed solution.	44
5.2	Illustration for image classification.	44
5.3	Segmentation in an image revealing different segments of food.	45
5.4	Estimating calories on food in an image.	47
5.5	Segmented image.	51
5.6	Difference between base value and predicted value.	53
5.7	Diagrams show the difference between original and predicted weight for two of the food categories.	55
5.8	Example image of milk.	56
1	Results for bread category.	72

2	Results for cheese category.	73
3	Results for chocolate milk category.	73
4	Results for crispbread category.	74
5	Results for milk category.	74
6	Results for yoghurt category.	75
7	Results for bread category.	76
8	Results for cheese category.	76
9	Results for chocolate milk category.	77
10	Results for crispbread category.	77
11	Results for milk category.	78
12	Results for yoghurt category.	78

List of Tables

2.1	Equations of some activation functions.	9
2.2	Macronutrients	21
2.3	A food table example [31].	23
3.1	Existing literature on food recognition and classification.	30
3.2	Existing literature on food segmentation.	33
3.3	Existing literature on weight and calorie estimation.	36
5.1	The results of top-1, top-3 and top-5 classification accuracy for Food-101 dataset.	49
5.2	The results of top-1, top-3 and top-5 classification accuracy for Food-11 dataset.	49
5.3	Comparing our results against previous results.	50
5.4	Confidence for each category.	52
5.5	IoU for each category.	54
5.6	Compare proposed solution against existing literature on food segmentation.	54
5.7	Result showing the accuracy of difference between real weight and estimated weight.	57
5.8	Sample results for four different food items.	58
5.9	Comparing our result against previous results.	58

Chapter 1

Introduction

Food is an essential part of everyday life. What we eat can have a big impact on the weight, and according to World Health Organization (WHO) the overweight and obesity has risen drastically the last 40 years. In 2016 more than 1.9 billion adults worldwide were overweight, where over 650 million of these were obese. If we consider the statistics of Norway for the year 2017, Norwegian Institute of Public Health (NIPH) stated in a report last updated late 2017 that 1 out of 4 men and 1 out of 5 women are obese. This is a huge concern and is also a big risk factor for diseases such as diabetes, heart disease, musculoskeletal disorders and some different type of cancers [1, 2].

Artificial Intelligence (AI) can be used to achieve a more healthy lifestyle. By using advanced networks such as ResNet, food can be recognized for what it is. Segmentation can be used to divide the image into smaller parts of the image. This thesis are using a segmentation method called You Only Look Once (YOLO), which finds all relevant elements in the images. The segmented food elements can then be used to calculate the weight of food in the image by sending the image into an inception network. This proposed solution is the first example of how estimation of grams can be achieved by using only a single image. The results from this thesis show that the network is able to predict the weight, and have a standard error of 8.95 for all categories and 2.40 for the best category which is bread.

1.1 Motivation and Problem Statement

Deep learning can be utilized for classifying things in an image or a video, recognize patterns etc. Thus, the main objective for this thesis is to use deep learning to classify the food in the image. The image should then be segmented to gather only the information regarding the food. Then the segmented parts is to be analyzed to gather nutrition based on the food in the image by finding out the size of the food.

Food classification has been done many times already, but there are still no well known method to figure out the nutrition levels of the food found in the images. One of the big issues regarding finding the nutrition levels in an image is that a image is two dimensional, and thereby difficult to figure out size of the food in the image. It will therefore be important to find a way to measure the amount of food found in the image.

There will also be a difference in how the food has been prepared. It will be easier to figure out the nutrition levels in an apple than in a lasagna. Complex dishes like lasagna can be difficult to figure out as a lot of the ingredients will be hidden in the image, and is therefore close to impossible to figure out. However an estimate can be made based on a known recipe for these type of dishes, giving it a close match. For simpler food elements it should be easier to figure out the nutrition levels as the image will be able to show everything included in the image.

To achieve the primary objective in this thesis, a version 3 inception network will be trained based on the segmented images and the given weight of the food will be sent in to the training network.

1.2 Research Questions

In this section, we discuss the research questions this thesis makes an effort to answer.

1. *Are Deep Neural Network (DNN) classification models suitable for classifying food images?*

To answer this research question, in the thesis we will be using a classic neural network called ResNet in order to get the classifications from food images. By using a pre-trained network it should be possible to only retrain the last layer of the network to match the classification model against food images.

2. *Is image segmentation a good strategy for counting calories in food images, and how can this be implemented?*

This question can be answered by looking at how the focus in the image can shift from the entire image to only the area containing food, which essentially will help in the prediction of food calories in the image. Image segmentation is implemented in the thesis by using a YOLO network called Darkflow.

3. *How can machine learning (ML) be trained to determine the weight of food in an image?*

This question can be answered by using an inception network which will learn to understand the weight based on the image segment from the segmentation part.

1.3 Thesis Goals

1. Examine the state-of-the-art research within the field of food recognition and how to classify food from the images.
2. Create a ConvNet that is able to classify food in the images.
3. Explore different techniques that can be used to calculate calories from the food in the images.
4. Implement a technique to find the calories in the image based on the research in goal 3.

1.4 Assumptions and Limitations

The assumptions and limitations are added as an aid to clarify why some of the choices have been made. By including assumptions, the reader will get a clear view of what is stated as the truth in the report. The limitations will help to limit the magnitude of the task at hand, and make it clear what is expected.

1.4.1 Assumptions

1. When working with food, there are some differences in how much each type of food weighs. For instance, a piece of bread can vary a lot in terms of weight based on the ingredients used to make the bread. Some low carbs bread is quite compact and heavy for their size in contrast to a wheat bread. An assumption has therefore been made that food within the same class will have the same weight for the same volume.
2. More complex dishes like lasagne and pizza as well as different kinds of soup are difficult to estimate as these dishes do not show the ingredients in a very good way. People are making these kind of dishes in different ways, and the ingredients will therefore vary a lot. This issue will be solved by assuming the dishes are the same as specified in the food table.

1.4.2 Limitations

1. When working with machine learning (ML), there is need for a lot of computational power. Training new models is quite computational intensive, and big companies use huge data centres for such tasks. One solution for this is to use pre-trained networks, but it will still be necessary to do some training. By using a GPU on a laptop or desktop this training takes time. It will therefore be a limitation in how good the training models will be, and can possibly give a lower accuracy than what could be expected.
2. The food size can be hard to predict as it is difficult to find out how close the food are in an image. By placing an known object like a ruler in the food images, it will be easier to estimate the size of the food.
3. In addition to the limitations listed above, time can also be looked at as a limitation. The project at hand is set to last one semester, which limits what tasks will be achievable during this time period.

1.5 Contributions

This thesis contributes with a proof of concept regarding finding calories of food from an image. Others have done similar tasks before, by either using two images from different positions of the food to estimate the size and others have looked at known textures to find the size of the food. In this thesis we estimate the weight of food in one single image by training a network on known weights for the food. This means that we will be able to predict the weight of food by only looking at one single image. The calories can then be calculated by using the weight together with information from a food table. This thesis also contributes with a new dataset containing weight for each food element in the images.

1.6 Thesis Outline

- **Chapter 2:** This chapter provides the background information needed in order to understand the task at hand.
- **Chapter 3:** This chapter presents the literature review for the previous work done in the field of machine learning for classifying food images and calorie estimation.
- **Chapter 4:** This chapter explains the pre-processing techniques that are used before feeding the data into neural network for the classification, segmentation and calorie count estimation.
- **Chapter 5:** This chapter provides the details of experimental setup that are used for the implementation of the different networks and also describes the experimental results we have achieved and the discussion of the results.
- **Chapter 6:** This chapter concludes the thesis by providing the summary of the work done in the thesis. Also, outline the potential work that could be done to our research in the future to achieve better desirable results.

Chapter 2

Background

This section starts with outlining the background theory and algorithms that have been used to answer the research questions of this thesis. Further, this chapter also presents the background information related to diet which includes nutrition, calories and food table.

2.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a network loosely based on how neurons of the human brain works. The ANN is more a framework than an algorithm and is used in many different ML algorithms. The network is a computing system that process complex data and translate it into something the computer understands. The ANN has no programmed task-specific rules in the algorithm, and will be able to predict a result based on examples [3]. In image recognition the ANN might be used to learn to identify people faces in the images based on lots of example data. The results from these training images can then be used to recognize faces in other images as the network has learned what it should be looking for.

2.1.1 Perceptron

The ANN is based on the perceptron which is a simple version of how the biology neuron works. Figure 2.1 show how a single perceptron look. When the neuron get the input, it will start multiplying this by a weight. This weight can later be adjusted according to how the error rate of each training is acting. For three inputs it will have three individual weights, one for each input. These values will then be multiplied together to get one value. In addition to the input and weight, there will also be an offset which is referred to as bias.

$$Y = \sum (weight * input) + bias \quad (2.1)$$

At first the weight and bias are chosen randomly, but for each iteration of training the values are getting closer and closer to an expected result. At the end of the process, the value are forwarded to a activation function which decides if it should be fired or not. There are many different forms of activation function, which handles the value differently [4, 5].

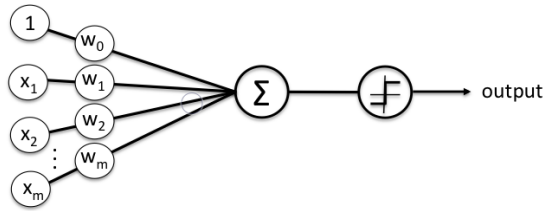


Figure 2.1: A single perceptron [6].

2.1.2 Activation Functions

The activation function calculates the sum and bias and check if the neuron should be fired as briefly mentioned in 2.1.1. The neuron has no clue of what the value should be, and can therefore be anything. In order to know when the neuron should fire and not, it is essential that there is an activation function which will activate and fire the neuron when the requirements in the activation function has been met. Table 2.1 show some of the different activation functions that exist, and how the equation for each of these are. For the perceptron in figure 2.1 a binary activation function has been used, which means that if the value from the perceptron is one or above it should fire. If it is below one it should not fire [5].

Name	Equation
Rectified Linear Unit (ReLU)	$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{otherwise} \end{cases}$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
TanH	$\tanh(x) = \frac{2}{1+e^{-2x}}$
Softmax	$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \text{ for } j = 1 \dots K$
Binary	$f(x) = \begin{cases} 1, & \text{if } w * x + b > 0 \\ 0, & \text{otherwise} \end{cases}$

Table 2.1: Equations of some activation functions.

2.1.3 Optimization

In order to optimize the ANN, the weights and bias has to be updated. During the optimization process an error function is defined, which calculates the error/cost value of the network at the output layer. The error value show the difference between the predicted value and the real value. There are different optimization versions to implement, where back-propagation is one of the most widely used optimization for ANN. When the network gets an error value, this will be back-propagated through the network until all the neurons have got the error value. Each neuron can then look to see if the neuron contributed in a positive or negative way, and can then adjust the weights by using gradient descent [7, 8].

2.1.4 Deep Neural Network

A deep neural network (DNN) is a larger network which have one or more inputs, one or more outputs and more than one hidden layer. These networks are based on multiple neurons working together, where each neuron is connected to each neuron in the next layer.

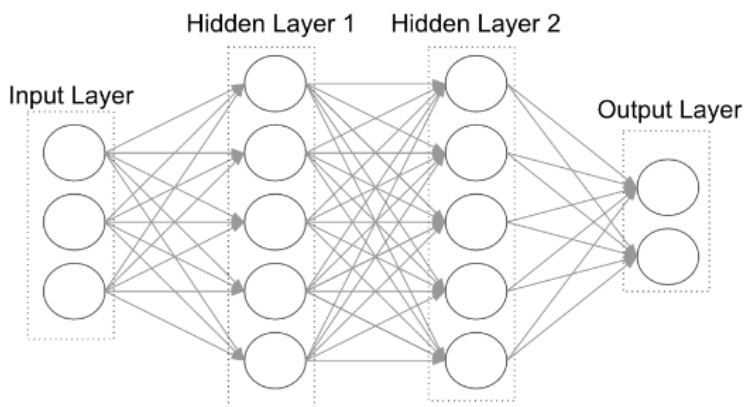


Figure 2.2: A DNN with two hidden layers [9].

2.2 Convolutional Neural Networks

Convolutional Neural Networks (ConvNet) has proven to be very useful when working with images as the network help to reduce the number of parameters to work with when looking at an image. The ConvNet can be divided into five steps:

1. Convolution
2. Activation
3. Pooling
4. Flattening
5. Full connection

The full connection layer is basically an ANN network which is explained in chapter 2.1 and will therefore not be addressed in this section.

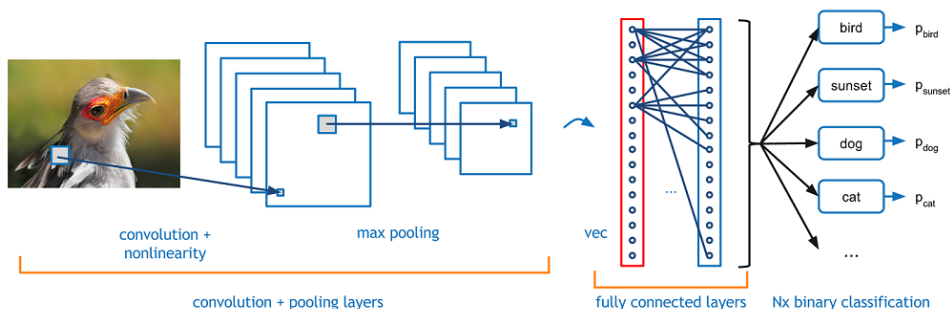


Figure 2.3: Overview over the ConvNet [10].

2.2.1 Convolution

The convolutional layer will look at the input image and try to figure out which elements in the image is worth looking at. This is done by using a sliding box which will go over each pixel of the image. The size of the sliding box as well as how big the steps are can vary. The sliding box will

be looking at the values of the image, so for a black and white image the values will be either 0 or 1, while a color image will be using 0 to 3 where three numbers decides the color. To find important features in the image, a feature detector is used. This filter will be the sliding box which will go over the image. For each step the feature detector will compare to the image matrix and find the numbers that match, and add the sum of these to the feature map. Figure 2.4 show how the lower left square of the input image only have one of number one compared to the feature detector which results in a one in the feature map as well [11].

The convolution layer also contribute by reducing the size of the matrix, which eventually will help speed up the process later. This is done by removing unnecessary information from the image giving the algorithm more room for working on the information that matters. A ConvNet is not limited to only have one convolution, and there can be created a feature map for different features in an image [11].

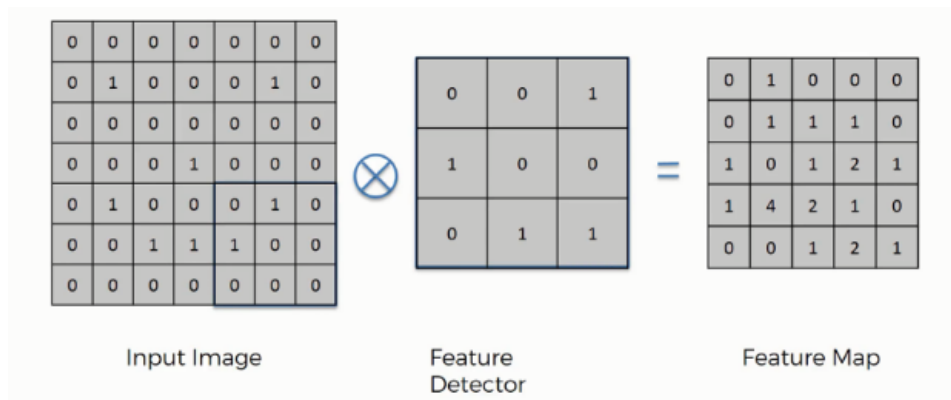


Figure 2.4: Converting an image into a feature map [12].

2.2.2 Pooling

Pooling is an optional part of ConvNet used in order for the algorithm to recognize an object in an image even if the object is placed in an odd angle. A dog should be able to be recognized even if the head of the dog is not aligned in the correct way or if the image resolution is not the same. Figure 2.5 show how max pooling works. Max pooling means that the highest number from the sliding box will be added to the pooled feature map.

Another option to max pooling would be average pooling which would find the average value from the sliding box at each step. From the figure we can see that the sliding box is a 2 x 2 with at stride of 2 which gives the results found in the pooled feature map. This way the important information in the image are preserved, and by going backwards on the pooling step it does not matter where in the square the number is set. For each feature map that is created, there will be a belonging pooling map [11].

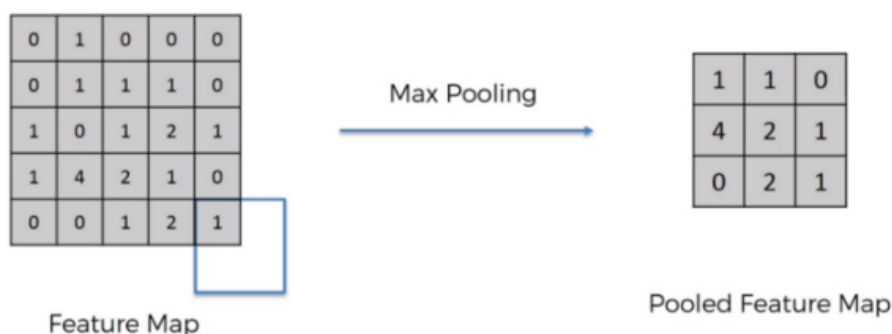


Figure 2.5: Add max pooling to the feature map [13].

2.2.3 Flattening

The flattening is a simple step which converts the pooled feature map into a column. This needs to be done in order for the ANN to use the feature map.

2.3 Residual Neural Network

The residual neural network (ResNet) is an advanced version of a ConvNet, which goes deeper than normal ConvNets. For a traditional ANN the accuracy will increase by adding more layers to the network, however at one point there will be a limit for the network to improve. Networks with a huge amount of layers will therefore start to experience some problems regarding the accuracy for the network, and will not be able to learn simple functions like an identity function. When the network got deep enough the accuracy would start to saturate and eventually start to degrade rapidly. This rapidly degradation is not a result of overfitting, but as a result of too many layers. ResNet tries to solve these problems by introducing a shortcut function [14].

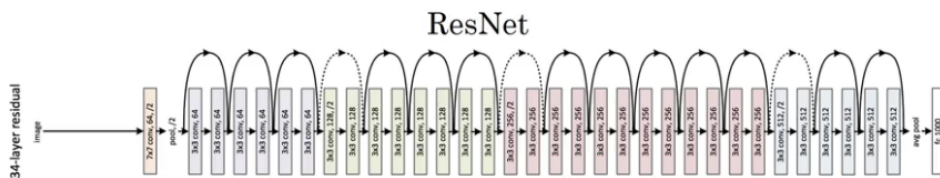


Figure 2.6: A 34-layer deep ResNet [14].

2.3.1 Residual Learning

Simply explained the residual is the rest of a task. In a network that would be quantity left when subtracting the input from the output like this: $F(x) = H(x) - x$. Turning this equation around gives the following equation which a residual block is defined by:

$$H(x) = F(x) + x \quad (2.2)$$

Here $H(x)$ are the output while x is the input for the residual block. $F(x)$ is the residual mapping to be learned. The residual block will try to learn what the output should be. As there is an identity connection in x , the layers will actually try to learn the residual $R(x)$. In a traditional network the network would try to learn the output $H(x)$, while in a residual network the network would try to learn the residual $R(x)$ [14, 15].

2.3.2 Residual Block

The difference from a normal ConvNet is that the network is divided into residual blocks which each has a shortcut through the block. As figure 2.7 show, the shortcut connection will be added together with the two new layers before the the ReLU for the last layer. By doing it this way, the identity can skip the new layers, which means that the network do not need to process all the data through every layer like a traditional network. When using the shortcut function the network is able to preserve more information across the layers, which gives each layer a better base. This means that the shortcut connection for the deeper layers should not have training error greater than more shallow networks [14, 15].

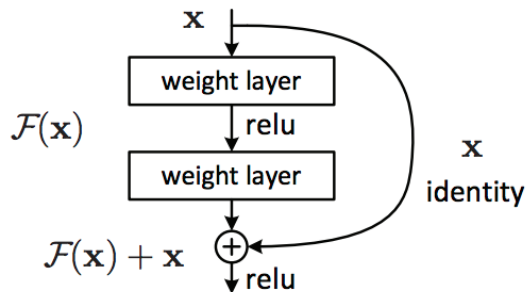


Figure 2.7: Residual block [14].

Each residual block will have the same dimension before the layers and after the layers. This is for the shortcut to have the same dimension as the network going through the layers. If for some reason the dimension should be different from each, this can be fixed by adding some extra padding to the matrix in order to get the same dimension through the residual block [14].

2.4 Image Segmentation

Image segmentation is a technique used to gather information from an image. By looking at the image, this technique is able to find objects in the image by dividing the image into either regions or categories. Each pixel will therefore be divided between these categories. If a pixel have similar greyscale of multiple values in the same category, forms a connected region and the pixels in different categories have different greyscale, this can be considered a good segmentation according to C. A. Glasbe *et al.* [16].

Segmentation can be viewed as a critical operation when handling images. The segmentation will start to recognize objects in the image instead of just the pixels. To get information from the image before segmentation, each pixel has to be looked at, while after the segmentation information can be gathered from the objects found in the image. Two ways to look at segmentation are region-based segmentation and edge-based segmentation [16].

2.4.1 Region-based Segmentation

Region-based segmentation is about grouping pixels together, where one important principle is the value similarity. This means that in order for a pixel to belong to a region the value variance needs to be within a certain limit [17]. The region-based segmentation can be categorized into three methods:

- Methods that merge pixels
- Methods that split image into regions
- Methods that includes both of the above

There are many different algorithms for region-based segmentation, where two of them are region growing and the split and merge algorithm. Region growing is a technique which connect regions or sub-regions into larger regions. This way the region will continue to grow until there are some edges which are different than current region. One approach for this is pixel

aggregation which will start with a seed and start expand by looking at the neighboring pixels. If the neighboring pixels have similar properties the region will expand, otherwise it is a sign that there should be a border here. Figure 2.8 present graphically how the process from the seed to final result looks like based on the input image [17].



Figure 2.8: Show region growing [18].

The merge and split algorithm is based on two operations and will have a limit which is defined for the maximum variance in pixel value for the region. The first operation is the region splitting and will start by setting the entire image as one region. It will then split the image into four sub-regions if the variance exceeds the limit. This will continue through the image until there is no region that exceeds the limit. The second stage is region merging and will look for regions that share an edge. If some regions have a common edge, these will merge together as long as the variance is not exceeding the limit that has been set [16]. Figure 2.9 displays how the image is split into sub regions which is then merged together again to get the shape of the entire region.

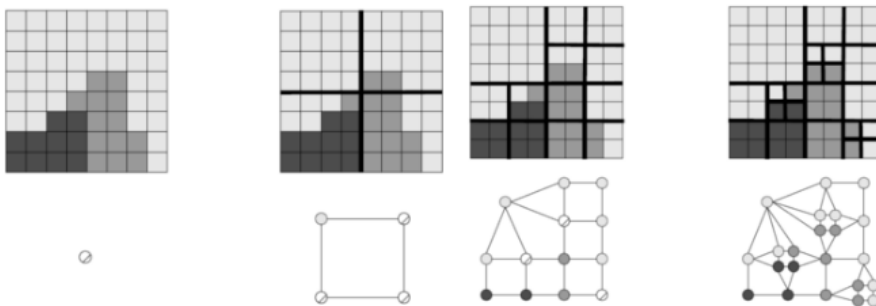


Figure 2.9: Example of region splitting in an image [18].

2.4.2 Edge-based Segmentation

Edge-based segmentation is another way to look at the segmentation. An edge in an image is the boundary where there is different grey level on each side of the boundary. In other words the edge is an area in the image where there is a significant change in contrast. To find the best edge, the image needs to be clear. Noisy and blurry images will help degrade the accuracy of the edge detection as the edges does not have the same clarity [19, 20].

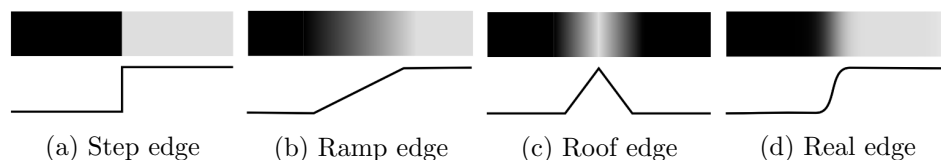


Figure 2.10: Different types of edges [20].

There are four main types of edges which are step edge, ramp edge, roof edge and real edge as shown in figure 2.10. The step edge is when there is a change in the intensity from one pixel to the next. This type of edge is not very common to find in images as the images are often more blurry. The ramp edge has a more gradient approach as figure 2.10b show, and is a closer match to the reality. The roof edge from figure 2.10c is when the edge changes and then goes back again. This is typically found if something like a stick is placed in front of a wall for instance. The real edge is probably the closest match to the reality. This model is the same as figure 2.10a, a step edge, with a Gaussian making the edge a bit blurry. When looking at an image the edges is often a bit blurry, and the real edge will therefore find these edges easier [20].

2.4.3 Segmentation with intersection over union

IoU is one of the most common metrics for comparing two shapes. This technique is also known as the Jaccard Index. IoU looks at the width, heights and location of the shapes and compare these to each other. It then calculates a value based on the areas of these shapes, and how much mismatch there are between the two shapes. This makes this good to evaluate segmentation and object detection. The following formula is used to achieve this:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2.3)$$

2.5 You Only Look Once

YOLO is an object detector. There are three main object detectors which are R-CNN with all its variants, Single Shot Detector (SSD) and YOLO. These object detectors have all different qualities to them and as figure 2.11 show, some gives better accuracy, while others provide better speed. According to the figure, YOLO is considered to be the fastest object detector, which has made it quite popular to use when looking for objects in live images.

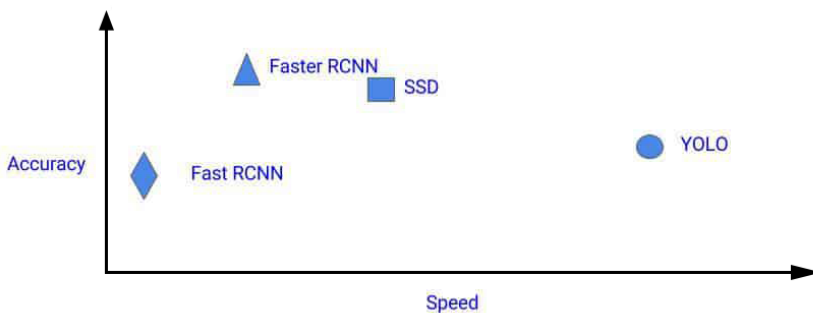


Figure 2.11: Illustrate the difference between segmentation methods.

In this algorithm, the network does not look at the entire image. Instead it takes parts of image which has high probabilities of consisting the object. In YOLO, one convolutional network anticipates the bounding boxes and the class probabilities for these boxes.

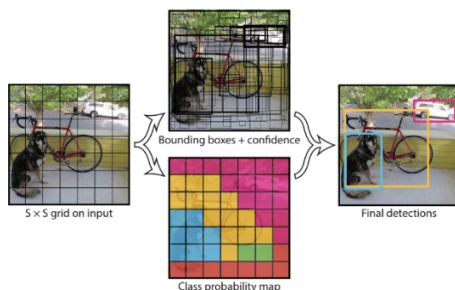


Figure 2.12: Show how YOLO works [21]

Image is divided into $m \times m$ grid and within each grid it is extracted n bounding boxes. For every bounding box, the network yields a class probability and offset values for the bounding box. when the class probability of bounding boxes is over the threshold value, then it will be selected and used to locate the object within the image. YOLO is faster than other object detection algorithms but struggles with small objects within an image.

2.6 Diet

Food is an essential part of everyone's life, and people are being more aware about what they are eating today than ever before [22]. To keep the body healthy it is important to have a varied diet as there is no food that contains all the important nutrition that is needed. A healthy diet should help us keep the right amount of energy throughout the day and help prevent some diseases that can follow a bad diet [23]. For instance a bad diet can lead to obesity, which again can lead to diseases like diabetes, hearth disease, musculoskeletal disorders, and some cancers [1]. Figure 2.13 indicates how a varied diet could look like, even though there are no definitive agreed upon food plate distribution.



Figure 2.13: How a varied diet could look like [24].

2.6.1 Nutrition

Nutrition is the study of nutrients in food and how this affects the body. Nutrition also looks at how nutrients can have an impact on the health and if there are some relationship between nutrition and diseases. Nutrition can be divided into two main categories which are macronutrients and micronutrients. Macronutrients is the type of nutrient that is needed in large amounts where some of them provides energy, and some do not. Table 2.2 includes the macronutrients and information about which of them provide energy.

Carbohydrates	Provides energy
Proteins	Provides energy
Fats	Provides energy
Fiber	Do not provide energy
Water	Do not provide energy

Table 2.2: Macronutrients

Micronutrients are the nutrients which are required in smaller amounts compared to the macronutrients. These nutrients are mainly vitamins and minerals that the body needs and only are able to get through food. The micronutrients are different for each type of food, and a varied diet is therefore recommended to get all the micronutrients. This will provide better health and help preventing diseases [25].

Vitamins are organic compounds which are needed in small amounts. Most of them has to come from food, while vitamin D is an example of a vitamin which comes primarily from the sun and not from food [26]. Minerals are inorganic substances needed for a variety of functions in the body to work and keep the body healthy. Some of the functions that requires the minerals are the bones, heart, brain and muscles. The minerals can be divided into macrominerals and mineral traces. Macrominerals are needed in larger amounts where some of the minerals are calcium and sodium. Mineral traces are needed in smaller amounts and some of the minerals here are iron, copper and zinc [25, 27, 28].

2.6.2 Calories

Calories are the measurement of the energy and is usually measured in kilocalories. Another unit of measurement for energy is kilojoules, 1 kcal = 4.2 kilojoules. This is just two ways to measure the energy, and both methods can be found on the food packing [29]. To keep the energy level high it is necessary to eat and drink throughout the day. The energy that is gained from consuming food and beverage is required in physical activity. When the amount of calories consumed is equal to the calories the body spend, the energy is in balance. If the calories spent are higher than the consumed calories, the body will convert the extra calories into fat which eventually will lead to weight gain. If the calories spent are less than the calories consumed, the body will get the extra calories from the fat reserve, which will lead to a weight reduction [23].

The amount of calories needed for a normal person depends on a number of criteria like gender and age. But still, an average estimate is considered to be around 2000 calories for women and 2500 calories for men each day [23].

2.6.3 Food Table

The food table is a table which includes the information about the food products. Most countries have their own table due to variation in food products from each country [30]. This means that food in a country can be found on the table for a given country. The table includes a detailed overview of all the nutrient values that can be found in each food item, and

the values are usual based on 100 grams of eatable food. Figure 2.3 gives an example on how a table looks like with the most important information.

Food Item	KJ	kcal	Fat	Carbo	Protein
Cheese, Norvegia	1458 KJ	351 kcal	27 g	0 g	27
Bread, coarse (75-100%)	896 KJ	212 kcal	2 g	32.5 g	12.4
Milk, whole	264 KJ	63 kcal	3.5 g	4.5 g	3.4 g

Table 2.3: A food table example [31].

Chapter 3

State of the Art

There is significant research that is going on in the areas of developing machine learning algorithms for food recognition, classification and calories estimation and counting food weight. In this chapter, we discuss some of the algorithms that have been developed by researchers in this field. In this thesis, our focus is to design such machine learning algorithms which can classify different foods.

3.1 Food Recognition and Classification

In the following section, we present the previous research that has been carried out by other researchers on food recognition and food images classification. The results of the previous research has been shown in table 3.1.

Taichi Joutou et al. [32] proposed a method called Multiple Kernel Learning MKL combined with a SVM, and can be considered as an extension to SVM. The method is applied on an unnamed dataset of 50 kinds of food categories where each image only show one dish. The result of this paper achieved an accuracy of 61.34% which is a big improvement compared to 34.64% which is the best result not using MKL. If three candidate categories were accepted the result improved to 80.05%. The researchers also tried lower resolution images taken by cellular-phones which gave an accuracy of 37.55%.

Taichi Joutou et al. [33] continued to build on the work that was done in [32] where MKL were used. This paper increase the number of categories in the dataset from 50 to 85 and also adding a new image feature called Histogram of Oriented Gradients (HoG). By doing these changes to the original paper, the researchers were able to get an accuracy of 62.52%. For the cellular-phone photos they were able to achieve an accuracy of 45.30%.

Zhimin Zong et al. [34] proposed a method using local binary pattern (LBP) for the local textural structure in an image. In addition, they represented the spartial information of the codewords by using shape context. The results were evaluated using the Pittsburg Fast-Food Image (PFI) dataset, and was compared against the two baseline classifiers which were Color Histogram + SVM classifier and Bag of SIFT features + SVM classifier. The researchers looked at six different categories where the proposed method was outperforming the two baselines in four of the six categories, and the color baseline was losing to the proposed method in every category. The results for this method varied between 52 and 82% accuracy.

Yuji Matsuda et al. [35] focused on images with multiple food dishes in each image. This paper looked at four different kinds of region detection methods to find the diferent food elements in an image: whole image, Deformable Part Model (DPM), Circle detector and the JSEG region segmentation. For image features, the researches looked at MKL, Bag-of-features with Spartial Pyramid (SP-BoF), HoG, Gabor Texture features and color histograms. To

evaluate candidate regions, a non-linear SVM was used, while a linear SVM was used to detect regions by the DPM. This method achieved an accuracy of 55.8% and 68.9% on the classification rate for multiple food-item images and single food-item images. For multiple food-item images this was an improvement of 40.4% compared to results without region detection.

Yoshiyuki Kawano et al. [36] looked at how deep ConvNets can improve the accuracy regarding food recognition. By using a pre-trained ConvNet network and only changing the last layer of the network, the network should be able to outperform conventional methods at the time such as HoG, BoF and Fisher Vector (FV). UEC FOOD 100 was the dataset that was used in this thesis. By combining the ConvNet with RootHoG-FV and color-FV, they were able to get an accuracy of 72.26% in the top-1 accuracy and 92% in the top-5 accuracy.

Lukas Bossard et al. [37] introduced the Food-101 dataset containing 101 categories with 1000 images each. The dataset is quite big compared to other dataset at the time, and is still one of the biggest public food datasets. This work also introduced a new method called Random Forest Discriminant Component (RFDC) which was based on Random Forest. In order to make a good comparison on Food-101 dataset, the researchers also implemented other state-of-the-art methods like Improved Fisher Vector (IFV), Bag-of-word Histogram (BOW), Mid-Level Discriminative Superpixels (MLDS) and ConvNet. On Food-101, the results for the RFDC outperforms the other state-of-the-art methods except for the ConvNet with 50.76% and 56.40%. RFDC was also giving competitive results on the MIT-Indoor dataset with 58.36% compared to the best, which is IFV + DMS with 66.87% accuracy.

Yoshiyuki Kawano et al. [38] presented how food recognition can be done locally on a smartphone without sending information to a server for processing. For better speed and accuracy HoG and FV had been considered over SIFT and BoF, together with a SVM classifier. The dataset used in this paper was the UEC FOOD-100. This method had been tested against a server-side food recognition system called Matsuda, and the results showed that the method provided in this paper had an improvement with the accuracy of 51.9% and 79.2% in the top-1 and top-5 accuracy. Matsuda on the other hand had an accuracy of 42.0% and 68.2% in the top-1 and top-5 accuracy.

Hamid Hassannejad et al. [39] used GoogLeNet and looked at the classification results for three different datasets: Food-101, UEC FOOD 100 and UEC FOOD 256. GoogLeNet was based on version 3 of the Inception Network. It is a network which contains 54 layers when compared to a normal ConvNet. This network had managed to increase the number of layers without overfitting and also reducing the computational power needed. The researchers used a pre-trained network created by Google and changed the architecture for the last layer to fit with new training. The biggest improvement by using GoogLeNet was found with the Food-101 dataset. The researchers obtained the accuracy of 88.28% and 96.88% in the top-1 and top-5 respectively. The UEC FOOD 100 and UEC FOOD 256 dataset had an top-1 accuracy of 81.45%, and 76.17%.

Niki Martinel et al. [40] proposed a solution called Wide-Slice Residual Network. This solution was a combination of two parts where one was slice convolutional layer which used to capture the vertical food structure, while the other part was a large residual learning architecture. The solution also used a larger number of feature maps for each convolution layer to handle problems regarding diminishing feature reuse. The method was tested on Food-101, UEC FOOD 100 and UEC FOOD 256. With this method, for Food 101 dataset, researchers obtained the accuracy of 90.27%, while UEC FOOD 100 and UEC FOOD 256 obtained an accuracy of 89.58% and 83.15%.

Paritosh Pandey et al. [41] created a method called Ensemble Net which consisted of 3 layers. The network used three different ConvNet networks that was connected as a Siamese network, meaning that each of the subnets were identical regarding configurations and weights. The three sub networks used were AlexNet, GoogLeNet and ResNet. The paper also contributes with a new dataset called Indian food image database which includes 50 classes of 100 images each. The method provided in this paper was compared with both the new Indian dataset as well as Food-101. The results showed that the Ensemble Net got an accuracy of 73.50% and 94.40% in the top-1 and top-5 accuracy for the Indian food dataset. GoogLeNet was the second best network on this dataset with 70.70% and 93.40% in top-1 and top-5 accuracy. For Food-101 the results were 72.12% and 91.61% in top-1 and top-5 accuracy.

Marc Bolaños et al [42] dealt with a classification problem where inception V3, ResNet 50 and ConvNet were used. This paper tries to recognize the food ingredients in an image. This was done by creating two new datasets, Ingredient-101 and Recipes5k. Ingredient-101 contained the most common ingredients from the Food-101 dataset, while Recipes5k contained 4826 unique recipes which each were an alternative way of preparing a dish in the Food-101 dataset. The results from this paper showed that by using the ingredient-101 dataset, ResNet50 gave the best result with a F1 score of 88.11%. For the Reciepe5k dataset the results showed that InceptionV3 was the best method with a F1 score of 47.51%.

Manal Chokr et al. [43] used Mathworks Image Processing Toolbox to extract raw features from food images and principal component analysis for dimensional reduction. After extracting the visual features from the image and performing feature reduction, each image was represented with a small number of features. Feature vector was passed to a classifier that outputs one of the six classes that has been used. Dataset used for this paper was the Pittsburgh fast-food image dataset and obtained an accuracy of 99.1% for the food classification.

Yanchao Liang et al. [44] used a Support Vector Machine (SVM) model to classify food types in general conditions. Faster R-CNN was used as deep convolutional network where it created bounding boxes. Image with RGB channel as input, the series of bounding boxes were produced. Every bounding box had the type of the food. Results from this paper showed that using Faster R-CNN, the researchers obtained the mean average precision with an accuracy of 93.0% for the food classification.

Patrick McAllister et al. [45] worked on classification of food by using two large DNN's, ResNet-152 and GoogLeNet to extract deep features from an image. The authors used six different food datasets i.e., Food-5k, Food-11 and Food-101. The author used four different classifier algorithms such as Gaussian Naive Bayes (GNB), SVM, ANN and Random Forest (RF). The results in this paper revealed some differences in accuracy between the different datasets and algorithms that had been used. The results showed that the SVM classifier combined with ResNet gave the best results for the Food-101 dataset with an accuracy of 64.98%. Food-5k and Food-11 both got the best results with ANN classifier combined with ResNet. Food-5k had an accuracy of 99.4% while Food-11 had an accuracy of 91.34%.

Author	Dataset	Accuracy (%)
Taichi Joutou et al. [32]	50 food categories	61.34
Taichi Joutou et al. [33]	85 food categories	62.52
Zhimin Zong et al. [34]	Pittsburg fast-food	52-82
Yuji Matsuda et al. [35]	100 food categories	55.80
Yoshiyuki Kawano et al. [36]	UEC FOOD 100	72.26
Lukas Bossard et al. [37]	Food-101	50.76
	MIT-indoor	58.36
Yoshiyuki Kawano et al. [38]	UEC FOOD 100	51.90
Hamid Hassannejad et al. [39]	Food-101	88.28
	UEC FOOD 100	81.45
	UEC FOOD 256	76.17
Niki Martinel et al. [40]	Food-101	90.27
	UEC FOOD 100	89.58
	UEC FOOD 256	83.15
Paritosh Pandey et al. [41]	Food-101	72.12
	Indian food image database	73.50
Marc Bolaños et al. [42]	Ingredient-101	88.11
	Reciepe5k	47.51
Manal Chokr et al. [43]	Pittsburg fast-food	99.10
Yanchao Liang et al. [44]	ECUSTFD	93.00
Patrick McAllister et al. [45]	Food-11	91.34
	Food-101	64.98
	Food-5k	99.40

Table 3.1: Existing literature on food recognition and classification.

3.2 Food Segmentation

In this section, we present the review on existing state-of-the-art related to food segmentation. Segmentation is different from classification because it is dividing an image into segments of more meaningful parts. The literature review is summarized and presented in table 3.2.

Marc Bosch et al [46] used a method based on generating multiple segmentation hypothesis. The methods that was used in this paper for the multiple segmentation hypothesis was Salient Region Detection, Multiscale segmentation and Fast rejection. The dataset had 32 food classes from 200 images where each image contained 6-7 food classes. The results varied from 11% to 98% over all the categories, but gave an average accuracy of 44%.

Ye He et al. [47] proposed an integrated segmentation and identification method by using the local variation for segmentation. The method started by segmenting the image, then the image was classified, before segment refinement was used to improve the result. The experiments that were done in this paper was based on 300 images from the Berkeley Segmentation Database. Since this paper proposed a method combining segmentation and classification, there were no results only for the segmentation part. The result for the segmentation combined with classification gave an accuracy of 34.0%.

Joachim Dehais et al. [48] used a pyramidal mean-shift together with a region growing algorithm. The input image was initially converted to CIELAB color space. Pyramid mean-shift filtering was then applied to reduce the details to decrease the computational cost. After this region growing and region merging was applied to find the segments. By using this method they were able to achieve a segmentation accuracy of 88.5%.

Keiji Yanai et al. [49] proposed a method on DCNN-based region detection. This method applied selective search to find bounding boxes and then perform back propagation over the DCNN for the bounding boxes. It then extracted the segments based on obtained saliency maps with GrabCut. The datasets used in this paper was UEC FOOD 100 and PASCAL VOC 2007, both of which have bounding boxes and class labels. The result gave an accuracy of 49.9% for the UEC FOOD 100 dataset, and 58.7% for the PASCAL VOC 2007 dataset.

Austin Myers et al. [50] used a system called DeepLab which is a deep learning model for semantic image segmentation. The model has been initialized on ImageNet and then finetuned on the Food201-segmented dataset. This a new dataset created for the segmentation part in this paper. The dataset have 201 classes of food with roughly 12,000 food images. The results from the segmentation gave an IoU accuracy between 0.19 and 0.25.

Joachim Dehais et al. [51] looked at segmentation by using a Region growing/merging method (explained in section 2.4.1). The paper proposes one automatic and one semi-automatic segmentation method. The automatic method combined the regional growing/merging with a deep ConvNet to detect food borders. The semi-automatic method required minimal input from the user by giving the seeds. The dataset used in paper consisted of 821 meal images, where each image contained one or multiple food items. The results from the automatic method gave an accuracy of 87.6% while the semi-automatic method gave an accuracy of 92.2%.

Yanchao Liang et al. [44] used Faster R-CNN and GrabCut to segment the food objects. GrabCut is an approach based on optimization by graph cuts. The method was implemented as a fully automatic segmentation to avoid any manual labeling. The method provides a precise contour of the food for each bounding box. This paper used a new dataset called ECUSTFD which contains 19 kinds of food where each image had a top view and a side view of the food element. The researchers did not include any results for the segmentation part of the process in this paper.

Author	Method	Dataset	Accuracy
Marc Bosch et al [46]	Multiple segmentation hypothesis	200 food images	44.0
Ye He et al. [47]	Local Variation	Berkeley Segmentation Db	-
Joachim Dehais et al. [48]	Mean-shift	-	88.5
Keiji Yanai et al. [49]	DCNN-based region detection	UEC FOOD 100 PASCAL VOC 2007	49.9 58.7
Austin Myers et al. [50]	Semantic Image Segmentation	Food201-segmented	19.0 - 25.0
Joachim Dehais et al. [51]	Region growing/merging	821 meal images	87.6
Yanchao Liang et al. [44]	Faster R-CNN	ECUSTFD	-

Table 3.2: Existing literature on food segmentation.

3.3 Weight and Calorie Estimation

In this section, we present the review on existing state-of-the-art related to finding weight and calories in the food images. This is different from classification and segmentation because weight and calorie estimation will try to find the size of food in an image. The literature review is summarized and presented in table 3.3.

Ye He et al. [47] used a method that combines the segmentation and classification by using a segmentation refinement step where feedback from classifier can be used to improve the segmentation. One of the goals for this paper was to be able to estimate food consumed based on only one image. In order to do this, two methods of estimating weight was suggested. The first method was when shapes are regular. A template for food specific shape was then used to estimate the volume. The second method was when shapes are not regular. This used a more direct area-based weight estimation.

Parisa Pouladzadeh et al. [52] developed a system to measure calories in images taken by a smartphone. The proposed method measures the volume of food in an image and estimates calories based on a nutrition table. The food was found in an image using segmentation which both look at color and texture and from this, different features were extracted. A SVM method was then used as a classifier to identify the food. Their solution for measuring volume was based on taking two images, one from the top and one from the side of the dish, and they also used a finger in both images to get the correct size. Results from the food recognition gave an accuracy of 92.21% by combining all features. The weight measurement results got an average accuracy of 86% for all food categories. For the calorie estimation, each category was measured against the real calorie value and a standard error was given based on the difference. Ten of the eleven categories gave a standard error between 0.07 and 0.65 while the last category achieved with an error rate of 2.97.

Austin Myers et al. [50] used deep learning algorithms like ConvNet and segmentation to recognize food items and predict the nutritional contents meals from images. Segmentation contribute to figure out their volume with the help of knowing the surface height of the food above the plate. They manipulated dataset into dataset-multilevel which are images from 23 different restaurants. The trained classifier used to map image to label.

Because of different type of food on the plate, Multi-label classifier was applied instead of multi-class classifier. The next step was to use some mathematical calculation to get the amount of calories. Results regarding the calorie estimation has not been presented in this paper, but for the volume estimation they was able to get an average relative error of 0.18 meters on the test set.

Manual Chokr et al. [43] extracted raw features from food images by using Mathworks Image Processing Toolbox. Every image was cropped and brightened and then re-sized to 4 % of its initial size. The next step was that image passed through the toolbox which returns three different matrices (RGB matrices). One matrix contains a cell for each pixel in the image that represents the red intensity of that pixel. Equivalently, the second and third matrices contain a cell for each pixel that represent the green and blue intensity of that pixel, respectively. The features extracted and reduced were passed to a machine learning predictor which produces a predicted amount of calories in the food item. In order to predict calories, they trained multiple classifiers based on the training data such as the Multilayer Perceptron, Support Vector Machines and Random Forests. In addition, they compared their pipelined approach of predicting calories by taking into consideration the type and size of the food item to a plain baseline that only takes into realistic in real world-scenarios. This have give a result with a mean absolute error of only 0.0933 on the test data.

Yanchao Liang et al. [44] calculated the volume of the food in order to get the amount of calories that present within that food. To get the volume of the food, they took the reference from one Yuan coin. Two images were needed to use this method to calculate calories, one from the side and another from the top. To find out the calories, some mathematics formulas were applied such as mass. Results show that the mass error did not exceed $\pm 20\%$ for most food types. However there were a few food types that went as high as $\pm 33.5\%$ on the mass error.

Author	Dataset	Calculation method	Error
Ye He et al. [47]	BSD	-	-
Parisa Pouladzadeh et al. [52]	-	Standard error	2.97
Austin Myers et al. [50]	Food201-multilabel	Average relative error	0.18
Manal Chokr et al. [43]	Pittsburg fast-food	Mean absolute error	0.09
Yanchao Liang et al. [44]	ECUSTFD	Mass error	± 33.50

Table 3.3: Existing literature on weight and calorie estimation.

Chapter 4

Methodology

The datasets that have been used in this thesis are Food-101, Food-11 and FoodX. In the first subsection, the datasets used in this thesis will be discussed. The second subsection explains the pre-processing methods that have been used to refine the data before feeding the data into a deep learning model like YOLO.

4.1 Dataset

In order to recognize food as well as calories from food in the image, some sort of training data is needed. In this thesis, three different datasets have been used where two of them are publicly available and the last one is a newly created dataset.

4.1.1 Food-11

Food 11 is a dataset which contains over 16 000 food images divided into training, testing and validation set. The dataset contains 11 major food categories which are: Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles/Pasta, Rice, Seafood, Soup, and Vegetable/Fruit.

4.1.2 Food-101

Food-101 is a larger dataset which has 101 major categories based on the most popular dishes. Each of these categories have 1000 food images, which give the dataset a total of 101 000 images. The dataset has not been cleaned and can therefore contain images with wrong labels.

4.1.3 FoodX

A new dataset has been developed together with some nutritional experts at the university of Agder in Kristiansand, and will be the primary dataset for calorie estimation in this thesis. The dataset will be referred to as FoodX in this thesis. It includes images of food together with a ruler which can be used to find the proportions of the food and the weight of the food. In addition to the images, the dataset also provides a document with the nutrition content for the food on each image.

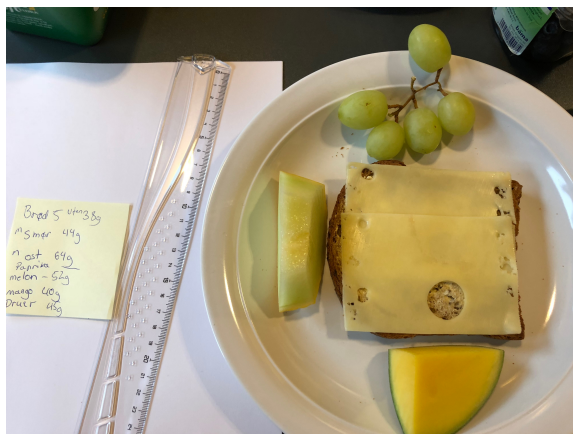


Figure 4.1: Example of how an image in the dataset look like, with 38g bread, 44g butter, 64g cheese, 52g melon, 40g mango, 45g grapes.

4.2 Data Pre-processing

In this subsection, the preparation of data in the datasets is presented.

4.2.1 Prepare Food-101

The Food-101 dataset consist of one folder containing all the categories. In order to use the dataset, there has to be some testing data. This has been done by taking 100 random images from each category and creating testing data from them. The dataset will then have a training set of 99 990 images and a testing set of 1100 images.

While Food-101 contains 101 categories of food, the Food-11 dataset only contains 11 categories of food. This creates an issue when the data is to be tested and needs to be addressed before testing the different methods. In order to test the dataset against a trained model of Food-11, some adjustments needs to be done. Each category in Food-101 is rearranged into the 11 categories that is found in Food-11. This way the Food-101 dataset will be able to be tested against the model created from Food-11.

4.2.2 Prepare FoodX

In order to use the new dataset, some pre-processing needs to be done to the dataset before using it. A label image software called labelImg [53] has been used to gather more information about the objects that can be found in an image. This is important as the training needs some information when training the network. This information can also be used to hide unnecessary objects in the images.

The label image software is used to draw boxes around the objects that can be important to further development. Important features for this thesis will be everything that has something to do with food as well as weight information found in an image. When the information about the objects has been marked, this will be stored in a XML file by using the PASCAL VOC format for storing the information. This information will then be used for segmentation as well as hiding information in an image. In order to test this dataset against another model, the dataset has to be adjusted in the same way as Food-101 by adding the images into the eleven categories of Food-11.

4.2.3 Bounding Box

All images in the FoodX dataset has some kind of note mentioning what type of food it is and the weight of the food. This could potentially be a problem when training on this dataset as the training could try to learn what kind of food it is and the weight of the food by recognizing the note instead of recognizing the food. To solve this potential problem, a black box has replaced the places where there are information about the food. This means that both the post-it notes and the weight display has been blacked out. Figure 4.2 gives an example of the two ways the weight is presented in the images.



Figure 4.2: Show how weight is presented in the images.

By getting the coordinates for the post-it and weight display in the image from the XML file, the image can be processed to hide this information by adding a black box over it. A function has been created to go over each of the XML files for the images and hide the post-it and weight. Figure 4.3 show how the image may look like after a post-it note has been blacked out.



Figure 4.3: Hiding a post-it note from the image.

Chapter 5

Experiments and Results

This section starts with presenting the experimental setup that we have used for classification, segmentation and weight estimation. Then we discuss the obtained results by conducting various experiments on the datasets.

5.1 Network Architecture

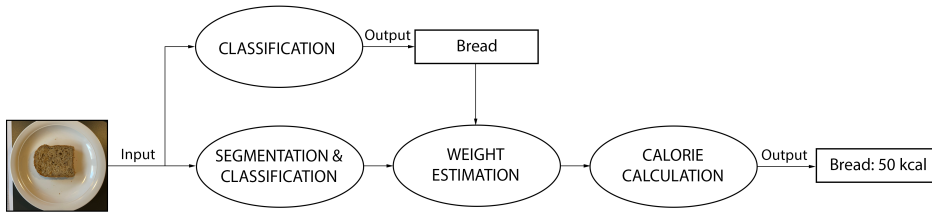


Figure 5.1: Pipeline showing the proposed solution.

This proposed architecture contains three main parts. The first part is about image classification, where we look at how food images has been classified in this solution. The second part is about the segmentation, this part explains how the food segments is found in the solution. The last part is calorie estimation, where the process of obtaining calories from food in an image is explained. Figure 5.1 show the pipeline for the proposed solution for the thesis.

5.1.1 Food Recognition and Classification

The second goal from section 1.3 is to create an image classifier that is able to recognize different food elements found in images. Image classification is a known problem within the topic of ML and many different techniques have been used to solve this problem.



Figure 5.2: Illustration for image classification.

In this proposed solution a network created by Google has been used. The network is based on the inception V4, and is called inceptionResNetV2. It

is the fourth version of inception network, but the second version of ResNet. This is a pre-trained network which has weights trained on ImageNet. In order to get a more specified network that suits our needs, some modifications has been applied to this network. The network is not specialized for food images, it is therefore something that we need to fix in order to get good results for the used datasets.

The solution for the network is to retrain the last layer of the network to work with the dataset used in this thesis. In additional to the base network, an extra fully connected layer and a logistic layer has been applied to the network. By creating a new last layer for the network, it is possible to change the amount of food categories. The new layer will be using a softmax activation layer which turns the output into a probability number. This number will be the accuracy of how good the food classification are performing. Training has been applied on both Food-11 and Food-101. FoodX contains too few images to train the network, and has therefore been excluded from the training.

5.1.2 Segmentation

Segmentation is done using a network called Darkflow. This is a network which is based on YOLO and is translated to work with the Tensorflow framework. The network saves the training progression at regular intervals which means that the training can be stopped at any given time. The network can then be tested or continue to train by using the saved checkpoints.

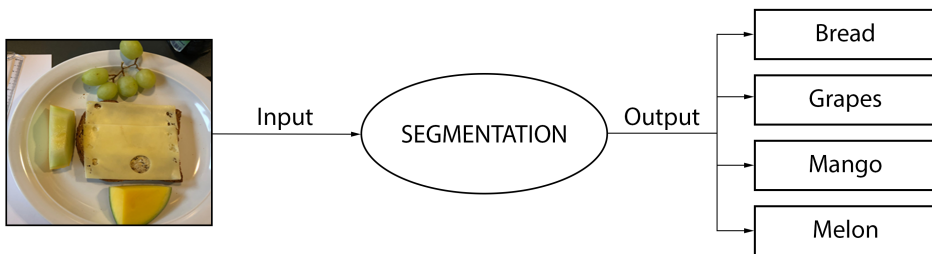


Figure 5.3: Segmentation in an image revealing different segments of food.

In order to train a network that works with a new dataset, some modification is necessary for the network to work properly. When Darkflow sees that a

weight is loaded, it will look for the configuration file that matches the weight file. When a new network is trained, Darkflow will compare the original configuration file with a new configuration file that includes the changes for the network. Darkflow will then include the weights for all layers which has no changes between the two configuration files. The layers that include changes will be the layers that will be retrained. It is therefore important to create a copy of the configuration file when creating a new network.

It is essential that the last two layers of the network is changed in the new configuration file to fit with a new dataset, and these layers will then have to be retrained with the new settings. The two last layers is one convolutional layer and one region layer. In each of these layers there is a variable that needs to be changed. In the region layer which is also the last layer of the network, the number of classes has to be set. By looking at the FoodX dataset, there is 26 classes which is all the food labels that was addressed in section 4.2.2. In the second to last layer which is the convolutional layer a filter variable has to be set according to the new dataset. This filter variable is calculated using the following formula: $n * (\text{classes} + 5)$. n is a variable in the region layer which is set to 5 by default. By using the FoodX dataset, the formula will look like this: $5 * (26 + 5) = 155$. When these variable has been changed, the network is ready to be trained.

To start the training this can either be done by using a terminal or by creating a python file doing the same thing. As the training is saved regularly and can be stopped at any given time, it does not matter how many epochs the training is set to.

Because the dataset has been labeled with 26 different classes, this can be used to recognize the food. When the segmentation is done training, the result should show the coordinates for a box surrounding each of the food elements in the image. Together with these coordinates is also a label mentioning what class these coordinates belongs to. Continuing forward, this information can be used when estimating weight in section 5.1.3.

In order to makes sure that the results provided for the segmentation are correct, IoU has been used (explained in section 2.4.3). By collecting the coordinates from the XML and JSON file which holds the coordinates for the correct and predicted coordinates of the food, the IoU would be able to show the error rate for the predicted coordinates.

5.1.3 Weight and Calorie Estimation

This section provides a proposed solution to the fourth goal in section 1.3. By looking at the size of the food, an inception network should be able to predict the weight of the food.

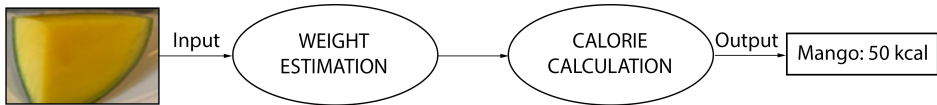


Figure 5.4: Estimating calories on food in an image.

The solution for this part are using the result from the previous section 5.1.2 where we are getting the segments from each food element. As figure 5.4 show, the input image is only one segment of food at the time. This is put into a inceptionV3 network, which is trained to estimate weight of the food in the segment. The predicted weight we get out will then be used to calculate the calories from this food element, and the output will be the name of the food together with calories.

When estimating weight of food, several things were done to test if there could be any improvements. First attempt were to take all the data in a convolutional net. The advantage of using all data might be that it can create more general weights for all data, however it does need to learn more, as different foods weigh differently for the same size. It may also look very different so the network will have to learn by using more data.

The test will be made by dividing data into several classes which are filtered down so that they will at minimum have 10 images in each class. In addition through segmentation a small patch of the image that contains the food will be extracted and this will be used to train on. This may make it easier for the network to recognize the food as the images is down-sampled during training and a small patch will therefore give a better resolution. However it might be more difficult for the network to pick up on other pieces of information like the size of the object compared to something known. An advantage is that the network will only have to train for a small subset of types of food at a time. It does however have less data, which could be an issue as the dataset is not that large to begin with.

When the network is done finding weight of the food, the results obtained are used to calculate the calories. This is done by using information from the food table in the document that follows the dataset. From the document it is possible to gather the calories per 100 gram as well as other nutritional information. By following the formula $\frac{\text{calories pr. 100g}}{100} * \text{weight} = \text{calories}$, we are able to calculate the estimated calories based on the estimated weight the inception network provided.

5.2 Results and Discussions

5.2.1 Food Recognition and Classification

Image recognition is a known task within AI and a lot of good results have been given by using ML techniques to handle the task. In this section, we will present the results of food image recognition by using the three datasets that are mentioned in section 4.1. Further, we discuss our results by comparing with the previous work. The experiments in this section have been conducted using two different models, one based on Food-101 and the other based on Food-11.

In the first experiment, we tested with and without fully connected layers at the end of ResNet to check whether there are any variations on the accuracies for three used datasets and the results show that we were able to achieve a small increase in the accuracy on Food-11 dataset. There was a small increase in the accuracy on FoodX and Food-101 datasets, when using more added layers. The model with fully connected layers have therefore been kept as this gave the best overall results.

In the second experiment, the number of epochs can have a huge impact on the result of the training. By re-training the networks a few times with different epoch settings have given some different results. For the Food-11 dataset, the result went from roughly 10% to above 90% by increasing the number of epochs from 15 to 25. The other datasets also had a boost, but not in the same scale as with the Food-11 dataset.

Dataset	Top-1 (%)	Top-3 (%)	Top-5 (%)
Food-101	72.38	79.21	80.30

Table 5.1: The results of top-1, top-3 and top-5 classification accuracy for Food-101 dataset.

The model which is based on Food-101 has only been tested against the Food-101 dataset as the other two dataset do not have enough classes to be tested against this model. Table 5.1 displays the results for the testing images from the Food-101 dataset. As we can see from this table, our model obtained 72.38% top-1, 79.21% top-3, 80.30% top-5 classification accuracies for food-101 dataset.

The model based on Food-11, is compared against all three datasets used in this thesis. Table 5.2 shows the results that have been presented for each of the datasets. Food-11 gives the best results in all three columns of the table, and results from the other datasets is not even close. The other datasets have been rearranged to fit into this model as mentioned in section 4.2.1 and as these datasets have different kind of images which affects the result. For FoodX, many of the images do not fit into any of the eleven categories that are found in the Food-11 dataset. This makes it difficult to predict these images and thereby gives the wrong output.

Dataset	Top-1 (%)	Top-3 (%)	Top-5 (%)
Food-101	49.51	60.99	67.33
Food-11	90.95	94.41	95.34
FoodX	31.19	62.71	75.25

Table 5.2: The results of top-1, top-3 and top-5 classification accuracy for Food-11 dataset.

In table 5.3 we see the results from previous papers compared to our results from this thesis marked in bold. Food-11 have a result close to the best result from other papers. For Food-101 we have achieved an accuracy of 72.38% which is among the best results for this dataset. However, the authors in [40] have obtained 90.27% on the same dataset. This may be because of the slice branch network which adds some more information in addition to

the ResNet. As FoodX is a new dataset, there is no paper to compare it against. Since FoodX is not a very big dataset, it has been tested against the Food-11 model and categories, rather than a model based on FoodX and its categories. This result has an accuracy of 31.19% and would likely be improved if a model with better categories for FoodX had been used.

Author	Dataset	Accuracy (%)
Taichi Joutou et al. [32]	50 food categories	61.34
Taichi Joutou et al. [33]	85 food categories	62.52
Zhimin Zong et al. [34]	Pittsburg fast-food	52-82
Yuji Matsuda et al. [35]	100 food categories	55.80
Yoshiyuki Kawano et al. [36]	UEC FOOD 100	72.26
Lukas Bossard et al. [37]	Food-101 MIT-indoor	50.76 58.36
Yoshiyuki Kawano et al. [38]	UEC FOOD 100	51.90
Hamid Hassannejad et al. [39]	Food-101 UEC FOOD 100 UEC FOOD 256	88.28 81.45 76.17
Niki Martinel et al. [40]	Food-101 UEC FOOD 100 UEC FOOD 256	90.27 89.58 83.15
Paritosh Pandey et al. [41]	Food-101 Indian food image database	72.12 73.50
Marc Bolaños et al. [42]	Ingredient-101 Reciepe5k	88.11 47.51
Manal Chokr et al. [43]	Pittsburg fast-food	99.10
Yanchao Liang et al. [44]	ECUSTFD	93.00
Patrick McAllister et al. [45]	Food-11 Food-101 Food-5k	91.34 64.98 99.40
Our work	Food-11 Food-101 FoodX	90.95 72.38 31.19

Table 5.3: Comparing our results against previous results.

5.2.2 Segmentation

Segmentation training has been conducted by using Darkflow (explained in section 5.1.2) and is based on YOLO (explained in section 2.5). The experiments regarding the segmentation have been done using the FoodX dataset. The dataset is not very big, which could be a variable when looking at the segmentation.

All test-images in the dataset is tested against the segmentation model. By doing this, the result will present itself by drawing boxes around each relevant element as shown in figure 5.5. In the below figure 5.5, it can be observed that, all the available food on the plate can be seen in the image, but the boxes are not set ideally around each food element. For instance, when detecting mango in the image there is some part that is not covered by the box. The mango box also includes too much of the image where there is no mango.

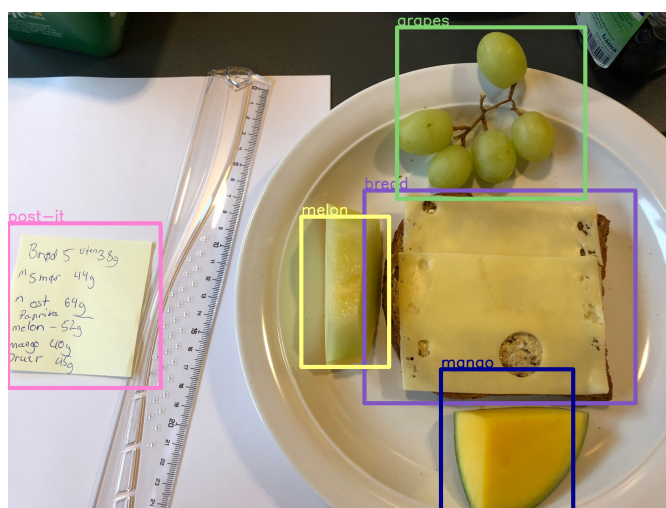


Figure 5.5: Segmented image.

Even though the boxes appear around the food, the segmentation is not certain to what it is on all the images. Table 5.4 gives an overview over the average confidence for each segmentation category. As table 5.4 shows, there are some variations in how good confidence there are among the categories. Post-it and weight is both above 80% confidence, while milk and bread gives

between 50-60% confidence. In the FoodX dataset, those categories are the most used categories. All the images in the dataset have either a post-it or a weight indicating the weight of the food. These categories have a lot of instances, and therefore the segmentation will be more confident with the result for these categories. The same reasoning can be made for the other categories and with more images the confidence will increase.

Category	Instances	Confidence
post-it	26	0.8123
Weight	18	0.9352
Milk	10	0.5940
bread	7	0.5086
cottagecheese	7	0.3129
cheese	1	0.1500
grapes	1	0.8000
mango	1	0.4100
melon	1	0.8200

Table 5.4: Confidence for each category.

Table 5.4 also shows that there are some categories with lower level of confidence like cheese and cottage cheese. Both of these categories have few images in the dataset hence the lower confidence. The segmentation was able to figure out all the categories on the test images except for one image where two slices of cheese upon each other was not marked with a box. By doing some research on why the cheese was not categorized, it was made clear that the reason for this was because the training set did not include any training images with two slices of cheese. Except from this one image, all the images were segmented with the correct categories, and even though the confidence is not at top at many of the images, it is still able to predict which category the segment belongs to, and that is the essential part of the segmentation process.

5.2.3 Intersection over Union

Based on the image in figure 5.5, it is obvious that the segmentation finds the target elements in the images, but it is difficult to see how good the

segmentation is just by looking at the image. It is therefore necessary with some kind of techniques to measure the segmentation results. A way to measure the result is by using a technique called IoU (explained in section 2.4.3) which compares the predicted results with boxes that were labeled during the preprocessing of the FoodX dataset. Figure 5.6 displays an example of how the result of this technique will look at an image. The green boxes display the original boxes that was drawn in labelImg while the red boxes display the predicted food segments.

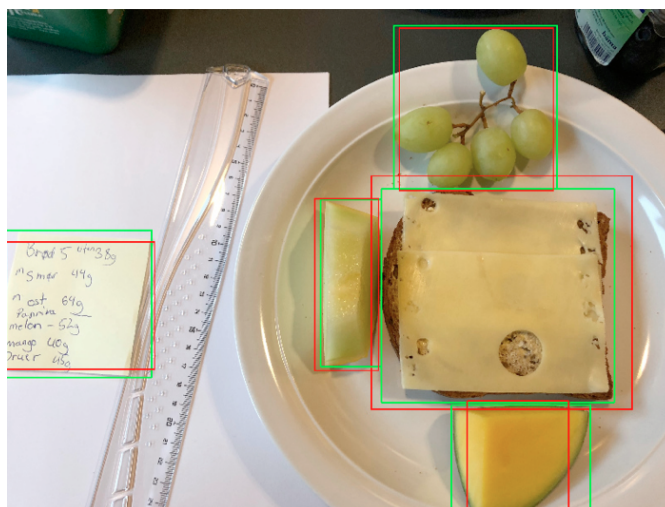


Figure 5.6: Difference between base value and predicted value.

By using IoU on all the test images it is possible to establish the accuracy for the segmentation. The accuracy is divided into classes that has been used for the segmentation. By dividing the results into categories, it became easy to see if some of the categories gives better results than other classes. Table 5.5 show the IoU accuracy for each category as well as how many times this category appears in the results. The table show that the IoU for each class is between 69-92% which is to be considered as a very good result for segmentation. The only exception is the cheese which has a lower IoU due to a problem with the segmentation on one of the images. By excluding images which do not have a segmentation, the IoU for cheese is also above 70%.

Category	Instances	IoU
Post-it	26	0.7823
Weight	18	0.7370
Milk	10	0.8043
Bread	7	0.7670
Cottagecheese	7	0.6957
Cheese	2	0.3779
Grapes	1	0.9196
Mango	1	0.7055
Melon	1	0.8412

Table 5.5: IoU for each category.

Author	Method	Dataset	Accuracy
Marc Bosch et al [46]	Multiple segmentation hypothesis	200 food images	44.0
Ye He et al. [47]	Local Variation	Berkeley Segmentation Db	-
Joachim Dehais et al. [48]	Mean-shift	-	88.5
Keiji Yanai et al. [49]	DCNN-based region detection	UEC FOOD 100 PASCAL VOC 2007	49.9 58.7
Austin Myers et al. [50]	Semantic Image Segmentation	Food201-segmented	19.0 - 25.0
Joachim Dehais et al. [51]	Region growing/merging	821 meal images	87.6
Yanchao Liang et al. [44]	Faster R-CNN	ECUSTFD	-
Our work	YOLO	FoodX	69.6 - 92.0

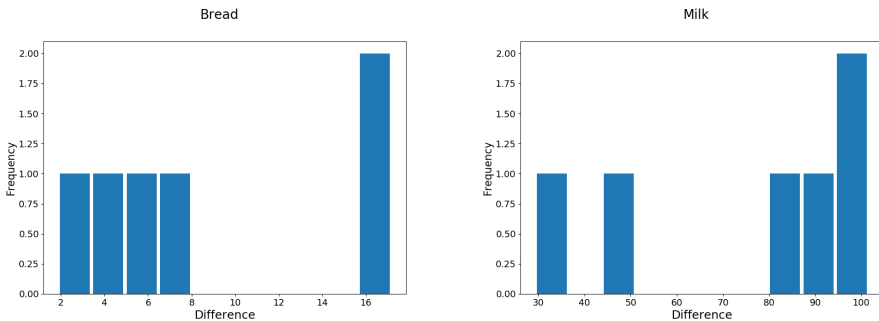
Table 5.6: Compare proposed solution against existing literature on food segmentation.

The results for segmentation and intersection over union would get better results if the dataset had been larger. The segmentation finds close to all the food in the images, and the IoU show that the accuracy for the boxes are quite good. This can therefore be considered a good result for this part of the thesis.

Table 5.6 shows how good our results are compared with previous papers using food segmentation. Our result from the experiments have left out the image that was not able to segment the cheese. This gives a result with an accuracy of 69.6%-92.0%. However, we cannot compare our results with the existing literature, as the datasets that have been used by other researchers are different from ours.

5.2.4 Weight and Calorie Estimation

The weight estimation has been performed using an Inception V3 network (explained in section 5.1.3). This section will present the results obtained from this network and discuss the findings.



(a) Results regarding bread.

(b) Results regarding milk.

Figure 5.7: Diagrams show the difference between original and predicted weight for two of the food categories.

The figure 5.7 demonstrates the results from the testing data for two of the categories, i.e. bread and milk. The Difference label represents the distance between the original and the estimated weight in grams and the Frequency label represents the number of images. By considering figure 5.7a, six testing images have been used to show how good the network is able to predict the weight of bread. The four bars on the left of the diagram show that four of the test images have a mass error of 2-8 grams, while the last two images shown in the bar on the right have a mass error of 16 grams. For all six images, the difference between the original weight and predicted weight is below 20 grams, but the best image has only an error of 2.5 gram from the original weight.

Figure 5.7b shows how the test images with milk is performing regarding the weight. In contradiction to figure 5.7a this diagram display a much worse result. One of the main reasons why the network is having trouble finding the correct weight for the milk is most likely because of the training images that have been used. Images of beverage has been taken from above as figure 5.8 shows. More diagrams showing the result from our findings can be found in appendix B.



Figure 5.8: Example image of milk.

The results from this network indicate that it is possible to get an accurate measurement of weight for food images. There can however be some problems for the network to understand the images in certain ways like the milk image in figure 5.8. By adding more images to the dataset with different

angles to the food would help the network to understand the amount of food in the image. If an image of a glass of milk was taken more from the side, the network would have better premises to see the height of the glass, and could perform better. By increasing the amount of data in the dataset, the network would likely be more precise in the prediction of the weights.

Category	Average Weight (grams)	Average Estimated weight	Average prediction difference from Fact	Standard Error
Bread	21.5	14.6	11.2	2.40
Milk	83.7	159.2	75.5	11.36
Cheese	21.0	8.1	14.4	-
Crispbread	18.0	12.6	9.5	5.36
Yoghurt	89.3	97.0	42.5	5.81
Chocolate milk	16.5	135.8	119.4	-

Table 5.7: Result showing the accuracy of difference between real weight and estimated weight.

Table 5.7 contains information of the original data in the FoodX dataset and for the predicted values as well as how they correlate. Average weight is calculated with $\sum^{images}(fact)$ which tells where the average for the prediction should be and also a bit of the scale of the numbers. Average Estimated Weight is the average of all predictions and is calculated with $\sum^{images}(prediction)$. It is to see if the average guess is lower or higher than what is correct. Average Prediction from Fact is calculated with $\sum^{images}|fact - prediction|$. Thus, it is possible to see how good each prediction is when compared to the fact. Ideally, this should be 0 as it would then estimate weight 100% correct. A value of 11.2 means that the prediction on average gets 11.2 grams wrong.

Table 5.7 also contains the standard error for each of the categories where the bread category has the lowest standard error with 2.40. Cheese and chocolate milk have too few testing images to get a correct standard error, but the standard error for all categories are 8.95.

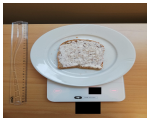

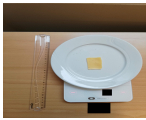

				
	Bread	Milk	Cheese	Crispbread
Actual	5 g 9.35 kcal	114 g 46.74 kcal	3 g 9.39 kcal	12 g 25.68 kcal
Predicted	10 g 17.96 kcal	159 g 65.19 kcal	8 g 25.0 kcal	13 g 27.8 kcal

Table 5.8: Sample results for four different food items.

Table 5.8 shows four sample images with different food items from the dataset. For each of these sample images, the predicted weight and calories are compared to the original weight and calories. The table displays the error margin that varies for all the sample results. For the crispbread the error is only 1 gram which gives an error of close to 2 calories. The milk sample on the other hand has an error of 45 grams which gives an error of 18 calories. The actual calories are gathered from the document with nutrition content that is attached to the dataset.

Author	Dataset	Calculation method	Error
Ye He et al. [47]	BSD	-	-
Parisa Pouladzadeh et al. [52]	-	Standard error	2.97
Austin Myers et al. [50]	Food201-multilabel	Average relative error	0.18
Manal Chokr et al. [43]	Pittsburg fast-food	Mean absolute error	0.09
Yanchao Liang et al. [44]	ECUSTFD	Mass error	± 33.50
Our work	FoodX	Standard error	8.95

Table 5.9: Comparing our result against previous results.

Table 5.9 displays the results from previous papers and compare the results against our result which is marked in bold. The other papers are all using different methods to calculate the error of their result, which makes it hard to compare our result with the other papers. Our result for the FoodX dataset use standard error as calculation method and have a result of 8.95 for all categories.

Chapter 6

Conclusion and Future Work

This chapter summarizes the achieved goals and results in the thesis. The future work will also be described for a better solution to the task.

6.1 Conclusion

In this thesis ¹ we propose an inception neural network to estimate the weight of food from single images. The main goal of this thesis is to use machine learning to classify an image, segment the food and determine weight and calories within an image. In addition, we introduce a new dataset called FoodX containing food pictures with corresponding weights.

By using a ResNet inception v4 network, we are able to train the network to recognize food in the images. Testing the network on Food-11 gives the best result regarding the image classification with an accuracy of 90.95% and 94.41% for the top-1 and top-3 respectively. Food-101 yields an accuracy of 49.51% and 60.99% for the top-1 and top-3 while FoodX has a top-1 and top-3 accuracy of 31.19% and 62.71%. A lot of the images in the FoodX dataset did not fit into the categories of Food-11 while testing accuracy and led to diminution of accuracy – making it an even more challenging dataset to work with than Food-11 and Food-101. The results obtained during testing confirm that deep neural networks are capable of classifying

¹We plan to submit this thesis as an academic paper at a later date

food images.

For the segmentation, we use a network called Darkflow which is based on YOLO. Results show that categories with more images has a higher confidence than categories with less images. The IoU show that the predicted segment box has an accuracy between 69% – 92%. These segments is then used to predict the weight of food. For the weight estimation, another inception network is used together with the FoodX dataset. By sending in a segment of a food picture, the network is able to predict a weight close to the original weight for some categories. The bread category is the best category with a standard error of 2.40, while the standard error for all categories is 8.95. As some of the categories have images taken from above, the network is having difficulty finding the correct weight.

A food table attached to FoodX is used to get the calories based on the estimated weight. More images and better angles for each category would help make the results better, but this thesis results indicates that ML can be used to determine the weight of food in an image.

6.2 Future Work

This section propose a future course of development for improvements to the current solution as well as new features to help improve the capability of the solution.

6.2.1 Increase Dataset

One of the biggest drawbacks with the FoodX dataset is lack of images. As the dataset includes information about the weight in the images, it is quite fitting for trying to estimate calories in food images. For future work it would therefore be important to increase number of images, which could provide a better foundation for the testing. With more images the solution could be tested on more categories of food and also potentially increase the accuracy of results.

6.2.2 Using Food Table API

The current solution is keeping the nutrition information in a file. It is only providing information regarding the images in the dataset, and would not be suitable for new food that is not found in the dataset. As a future work it would therefore be a good idea to look more at the national food table and gather information regarding the food from its API.

6.2.3 Use of Included Ruler

All pictures in the FoodX dataset include a ruler. This is placed so that an algorithm may use it to get better knowledge of the distance from the camera to the food. Since the ruler exists in the images during training, it is difficult to tell if information is preserved or used by the network. It may therefore be advantageous to extract information from the ruler by using the OpenCV library for Python, as it has some libraries for detecting size in an image. The information about size is then applied to the dense layer of the inceptionV3 network which can be done by using functional API in Keras. This may be either pixels per cm, or the size of the food in cm^2 .

Bibliography

- [1] World Health Organization. *Obesity and Overweight*. 2018.
- [2] NIPH. *Overweight and obesity in Norway*. Tech. rep. 2014. URL: <https://www.fhi.no/en/op/hin/lifestyle/overweight-and-obesity-in-norway---/>.
- [3] DeepAI. *Neural Network*. URL: <https://deepai.org/machine-learning-glossary-and-terms/neural-network>.
- [4] Christoph. *Perceptrons - the most basic form of a neural network*. 2016. URL: <https://appliedgo.net/perceptron/>.
- [5] Avinash Sharma. *Understanding Activation Functions in Neural Networks*. 2017. URL: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>.
- [6] Francisc Camillo. *Neural Representation of Logic Gates*. 2017. URL: <https://towardsdatascience.com/neural-representation-of-logic-gates-df044ec922bc>.
- [7] Mina Niknafs. *Neural Network Optimization*. Tech. rep. URL: http://courses.mai.liu.se/FU/MAI0083/Report_Mina_Nikanfs.pdf.
- [8] Anish Singh Walia. *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent*. 2017. URL: <https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f>.
- [9] Adil Moujahid. *A Practical Introduction to Deep Learning with Caffe and Python // Adil Moujahid // Data Analytics and more*. 2016. URL: <http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/>.

- [10] Deshpande Adit. *A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – CS Undergrad at UCLA ('19)*. 2016. DOI: [10.1097/MD.0b013e31822403e9](https://doi.org/10.1097/MD.0b013e31822403e9). URL: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>.
- [11] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. 2018. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [12] SuperDataScience Team. *Convolutional Neural Networks (CNN): Step 1- Convolution Operation - Blogs SuperDataScience - Big Data — Analytics Careers — Mentors — Success*. 2018. URL: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation>.
- [13] Choi Won-Joon. *[DL Udemy] 1–2. CNN Pooling & Flattening – Won-Joon Choi – Medium*. 2018. URL: <https://medium.com/@cwj5050/dl-udemy-2-pooling-flattening-b46ea460e0d0>.
- [14] Kaiming He et al. *Deep Residual Learning for Image Recognition*. Tech. rep. 2015. URL: <http://arxiv.org/abs/1512.03385>.
- [15] Sabyasachi Sahoo. *Residual blocks - Building blocks of ResNet*. 2018. URL: <https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec>.
- [16] C. A. Glasbey and G. W. Horgan. “Image Analysis for the Biological Sciences.” In: *Biometrics* (2006). ISSN: 0006341X. DOI: [10.2307/2533987](https://doi.org/10.2307/2533987).
- [17] Hassana Grema Kaganami and Zou Beiji. “Region-based segmentation versus edge detection”. In: *IIH-MSP 2009 - 2009 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. 2009. ISBN: 9780769537627. DOI: [10.1109/IIH-MSP.2009.13](https://doi.org/10.1109/IIH-MSP.2009.13).
- [18] Aurélie Bugeau. *Détection et segmentation d objets*. URL: <https://docplayer.fr/39326277-Detection-et-segmentation-d-objets.html>.
- [19] Larry S. Davis. “A survey of edge detection techniques”. In: *Computer Graphics and Image Processing* 4.3 (Sept. 1975), pp. 248–270. ISSN: 0146-664X. DOI: [10.1016/0146-664X\(75\)90012-X](https://doi.org/10.1016/0146-664X(75)90012-X). URL: <https://www.sciencedirect.com/science/article/pii/0146664X7590012X>.

- [20] Philippe Cattin. *Image Segmentation - Introduction to Signal and Image Processing*. 2016. URL: [https://miac.unibas.ch/SIP/07-Segmentation.html#\(22\)](https://miac.unibas.ch/SIP/07-Segmentation.html#(22)).
- [21] *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. URL: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>.
- [22] Timi Gustafson. *Younger Consumers Are More Health Conscious Than Previous Generations*. 2017. URL: https://www.huffingtonpost.ca/timi-gustafson/younger-consumers-are-mor_b_14290774.html.
- [23] Katie Grimwood. *A healthy balanced diet*. 2017. DOI: 10.12968/eqhe.2017.37.12. URL: <https://www.nutrition.org.uk/healthyliving/healthydiet/healthybalanceddiet.html>.
- [24] *Vinn-vinn: Mat for både deg og jorden – Kore*. URL: <http://www.kore.no/blogg/vinn-vinn-mat-for-bade-deg-og-jorden/>.
- [25] Lizzie Streit. *Micronutrients: Types, Functions, Benefits and More*. 2018. URL: <https://www.healthline.com/nutrition/micronutrients>.
- [26] Christian Nordqvist. *Vitamins: What are they and what do they do?* 2017. URL: <https://www.medicalnewstoday.com/articles/195878.php>.
- [27] *Minerals*. URL: <https://medlineplus.gov/minerals.html>.
- [28] My Basket et al. *Minerals and trace elements Attachments* : 2019. URL: <https://www.nutrition.org.uk/nutritionscience/nutrients-food-and-ingredients/minerals-and-trace-elements.html?showall=1&limitstart=>.
- [29] Ingrid van Heerden. *Calories, kilojoules: which is it?* 2008. URL: <https://www.health24.com/Diet-and-nutrition/Weight-loss/Calories-kilojoules-which-is-it-20120721>.
- [30] *matvaretabell*. URL: <https://sml.sn1.no/matvaretabell>.
- [31] *Matvaretabellen*. URL: <http://www.matportalen.no>.
- [32] Taichi Joutou and Keiji Yanai. “A food image recognition system with Multiple Kernel Learning”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, Nov. 2009, pp. 285–288. ISBN: 978-1-4244-5653-6. DOI: 10.1109/ICIP.2009.5413400. URL: <http://ieeexplore.ieee.org/document/5413400/>.

- [33] Hajime Hoashi, Taichi Joutou, and Keiji Yanai. “Image Recognition of 85 Food Categories by Feature Fusion”. In: *2010 IEEE International Symposium on Multimedia*. IEEE, Dec. 2010, pp. 296–301. ISBN: 978-1-4244-8672-4. DOI: [10.1109/ISM.2010.51](https://doi.org/10.1109/ISM.2010.51). URL: <http://ieeexplore.ieee.org/document/5693856/>.
- [34] Zhimin Zong et al. “On the Combination of Local Texture and Global Structure for Food Classification”. In: *2010 IEEE International Symposium on Multimedia*. IEEE, Dec. 2010, pp. 204–211. ISBN: 978-1-4244-8672-4. DOI: [10.1109/ISM.2010.37](https://doi.org/10.1109/ISM.2010.37). URL: <http://ieeexplore.ieee.org/document/5693842/>.
- [35] Yuji Matsuda, Hajime Hoashi, and Keiji Yanai. “Recognition of multiple-food images by detecting candidate regions”. In: *Proceedings - IEEE International Conference on Multimedia and Expo*. 2012. ISBN: 978-1-4673-1659-0. DOI: [10.1109/ICME.2012.157](https://doi.org/10.1109/ICME.2012.157).
- [36] Yoshiyuki Kawano and Keiji Yanai. “Food image recognition with deep convolutional features”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct*. 2014. ISBN: 9781450330473. DOI: [10.1145/2638728.2641339](https://doi.org/10.1145/2638728.2641339).
- [37] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. “Food-101 - Mining discriminative components with random forests”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2014. ISBN: 9783319105987. DOI: [10.1007/978-3-319-10599-4_{_}29](https://doi.org/10.1007/978-3-319-10599-4_{_}29). URL: https://www.vision.ee.ethz.ch/datasets_extra/food-101/static/bossard_eccv14_food-101.pdf.
- [38] Yoshiyuki Kawano and Keiji Yanai. “Automatic expansion of a food image dataset leveraging existing categories with domain adaptation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2015. ISBN: 9783319161983. DOI: [10.1007/978-3-319-16199-0_{_}1](https://doi.org/10.1007/978-3-319-16199-0_{_}1).
- [39] Hamid Hassannejad et al. “Food Image Recognition Using Very Deep Convolutional Networks”. In: *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa '16*. 2016. ISBN: 9781450345200. DOI: [10.1145/2986035.2986042](https://doi.org/10.1145/2986035.2986042).

- [40] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. “Wide-slice residual networks for food recognition”. In: *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*. 2016. ISBN: 9781538648865. DOI: [10.1109/WACV.2018.00068](https://doi.org/10.1109/WACV.2018.00068). URL: <https://arxiv.org/pdf/1612.06543.pdf>.
- [41] Paritosh Pandey et al. “FoodNet: Recognizing Foods Using Ensemble of Deep Networks”. In: *IEEE Signal Processing Letters* (2017). ISSN: 10709908. DOI: [10.1109/LSP.2017.2758862](https://doi.org/10.1109/LSP.2017.2758862). URL: <https://arxiv.org/pdf/1709.09429.pdf>.
- [42] Marc Bolaños, Aina Ferrà, and Petia Radeva. “Food Ingredients Recognition Through Multi-label Learning”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2017. ISBN: 9783319707419. DOI: [10.1007/978-3-319-70742-6_{_}37](https://doi.org/10.1007/978-3-319-70742-6_{_}37). URL: <https://arxiv.org/pdf/1707.08816.pdf>.
- [43] Manal Chokr and Shady Elbassuoni. *Calories Prediction from Food Images*. Tech. rep. American University of Beirut, 2017, pp. 4664–4669. URL: <https://www.aaai.org/ocs/index.php/IAAI/IAAI17/paper/viewFile/14204/13719>.
- [44] Yanchao Liang and Jianhua Li. “Deep Learning-Based Food Calorie Estimation Method in Dietary Assessment”. In: (June 2017). URL: <http://arxiv.org/abs/1706.04062>.
- [45] Patrick McAllister et al. “Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets”. In: *Computers in Biology and Medicine* (2018). ISSN: 18790534. DOI: [10.1016/j.combiomed.2018.02.008](https://doi.org/10.1016/j.combiomed.2018.02.008).
- [46] Fengqing Zhu, Marc Bosch, and Nitin Khanna. “Multilevel Segmentation for Food Classification in Dietary Assessment”. In: *Image and Signal Processing and Analysis* (2013). ISSN: 1878-5832. DOI: [10.1016/j.micinf.2011.07.011.Innate](https://doi.org/10.1016/j.micinf.2011.07.011.Innate).
- [47] Ye He et al. “Food Image Analysis: Segmentation, Identification and Weight Estimation.” In: *Proceedings. IEEE International Conference on Multimedia and Expo 2013 (July 2013)*. DOI: [10.1109/ICME.2013.6607548](https://doi.org/10.1109/ICME.2013.6607548). URL: <http://www.ncbi.nlm.nih.gov/pubmed/28572873>.

- [48] Marios Anthimopoulos et al. “Segmentation and recognition of multi-food meal images for carbohydrate counting”. In: *13th IEEE International Conference on BioInformatics and BioEngineering, IEEE BIBE 2013*. 2013. ISBN: 9781479931637. DOI: [10.1109/BIBE.2013.6701608](https://doi.org/10.1109/BIBE.2013.6701608). URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6701608>.
- [49] Wataru Shimoda and Keiji Yanai. *CNN-based food image segmentation without pixel-wise annotation*. Tech. rep. 2015, pp. 449–457. DOI: [10.1007/978-3-319-23222-5_{_}55](https://doi.org/10.1007/978-3-319-23222-5_{_}55). URL: <http://koen.me/research/selectivesearch/>.
- [50] Austin Myers et al. “Im2Calories: Towards an automated mobile vision food diary”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015. ISBN: 9781467383912. DOI: [10.1109/ICCV.2015.146](https://doi.org/10.1109/ICCV.2015.146). URL: https://www.cs.ubc.ca/~murphyk/Papers/im2calories_iccv15.pdf.
- [51] Joachim Dehais, Marios Anthimopoulos, and Stavroula Mougiakakou. “Food Image Segmentation for Dietary Assessment”. In: (2016), pp. 23–28. DOI: [10.1145/2986035.2986047](https://doi.org/10.1145/2986035.2986047). URL: <http://dx.doi.org/10.1145/2986035.2986047>.
- [52] Parisa Pouladzadeh, Shervin Shirmohammadi, and Rana Al-Maghrabi. *Measuring calorie and nutrition from food image*. Tech. rep. 8. 2014, pp. 1947–1956. DOI: [10.1109/TIM.2014.2303533](https://doi.org/10.1109/TIM.2014.2303533). URL: <https://www.site.uottawa.ca/~shervin/pubs/FoodRecognition-IEEE-TIM-final.pdf>.
- [53] Darrenl Tzutalin. *LabelImg*. 2015. URL: <https://github.com/tzutalin/labelImg>.

Appendices

A Code repository

Code and supplementary material are available at:
<https://github.com/Runari14/Weight-Estimation-ML>

B Weight results

This chapter show the results in a diagram for each of the six categories which have at least ten images in the dataset. The chapter is divided into two parts. The first part show the results based on the validation data and the second part show the results based on the training data. The diagrams shown in this chapter have the amount of images on the Y-axis and the weight difference between the original weight and the predicted weight in gram.

B.1 Results from Validation Set

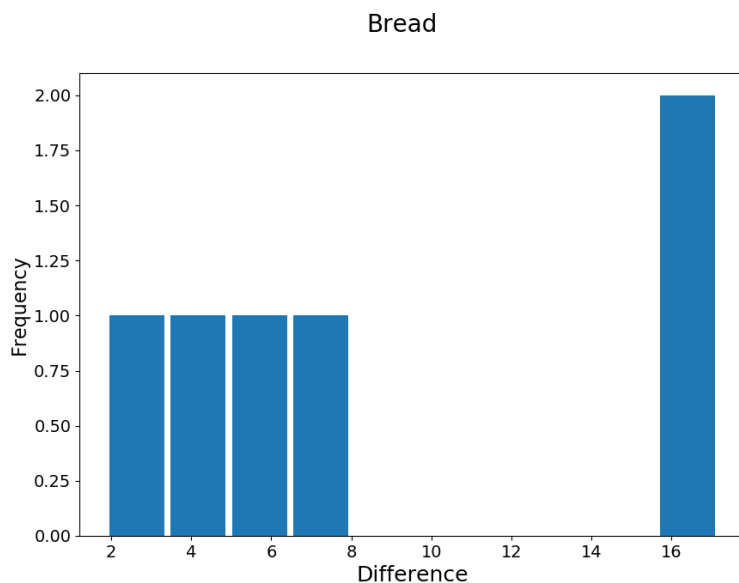


Figure 1: Results for bread category.

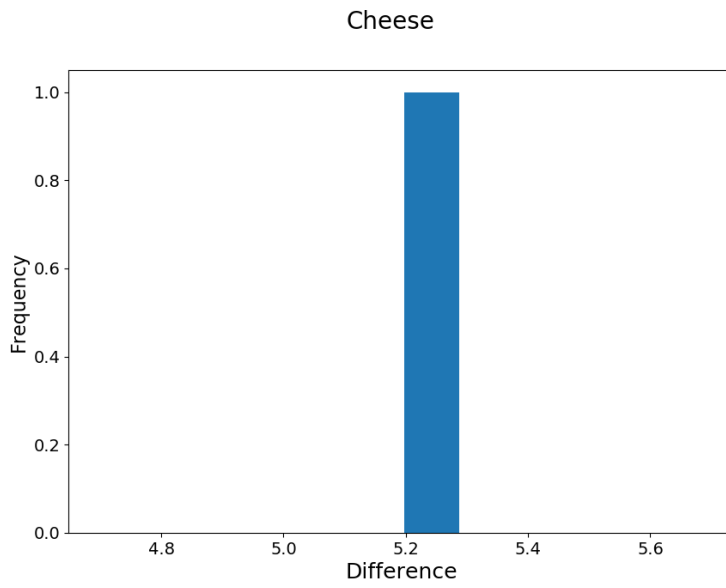


Figure 2: Results for cheese category.

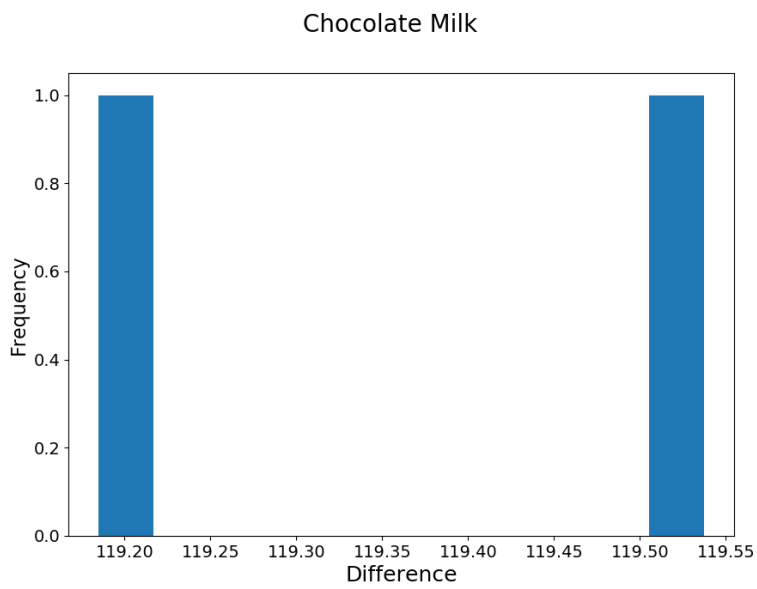


Figure 3: Results for chocolate milk category.

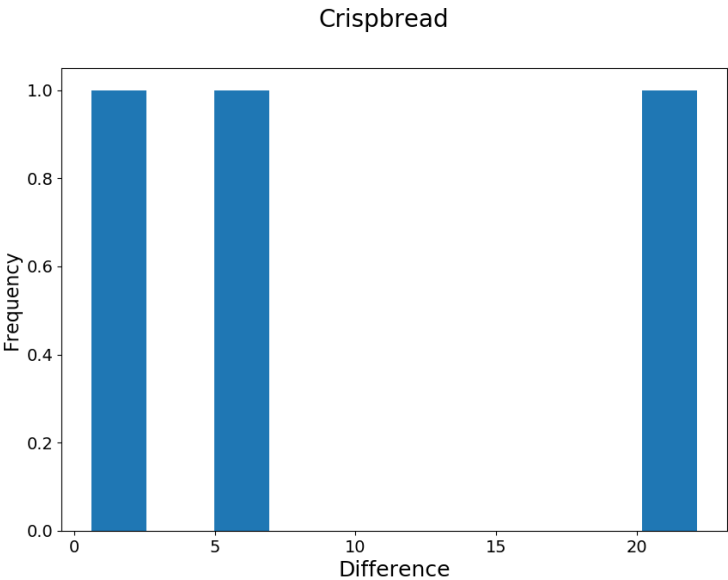


Figure 4: Results for crispbread category.

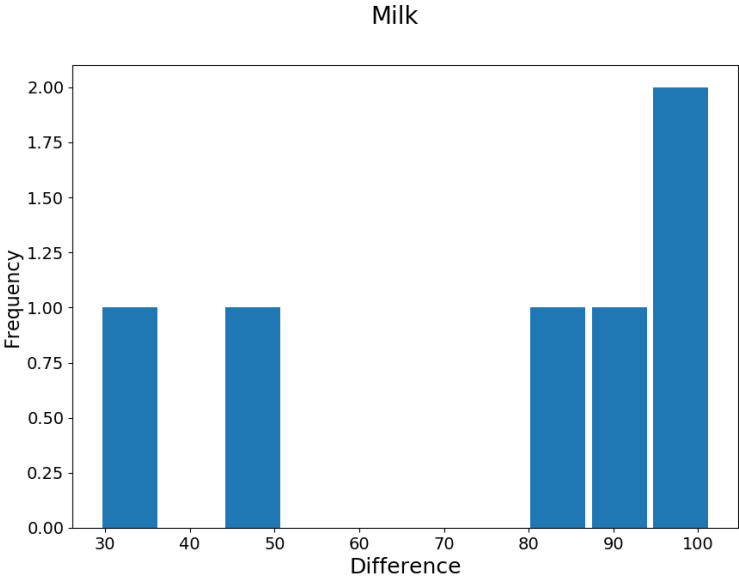


Figure 5: Results for milk category.

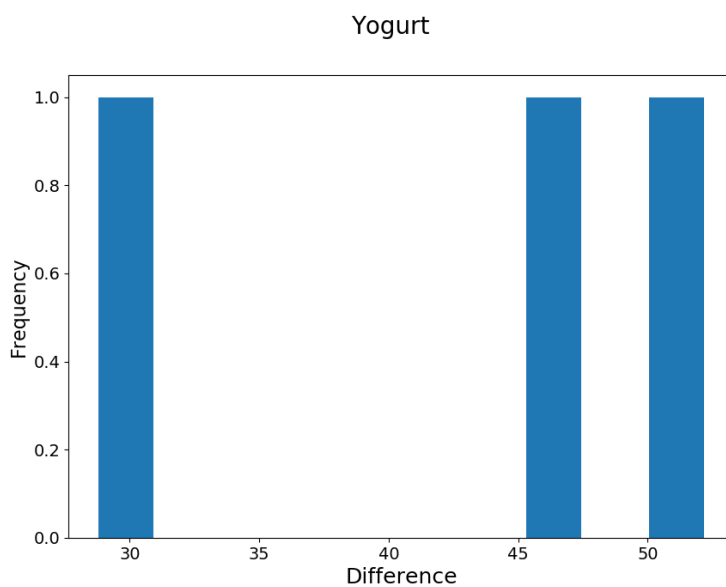


Figure 6: Results for yoghurt category.

B.2 Results from Training Set

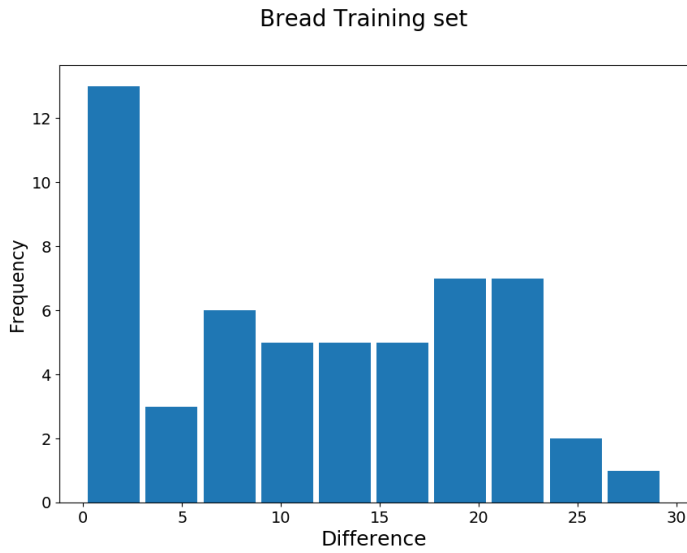


Figure 7: Results for bread category.

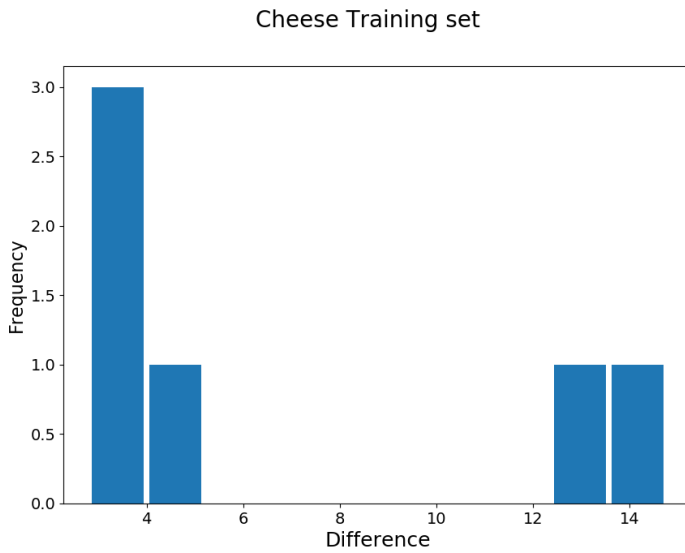


Figure 8: Results for cheese category.

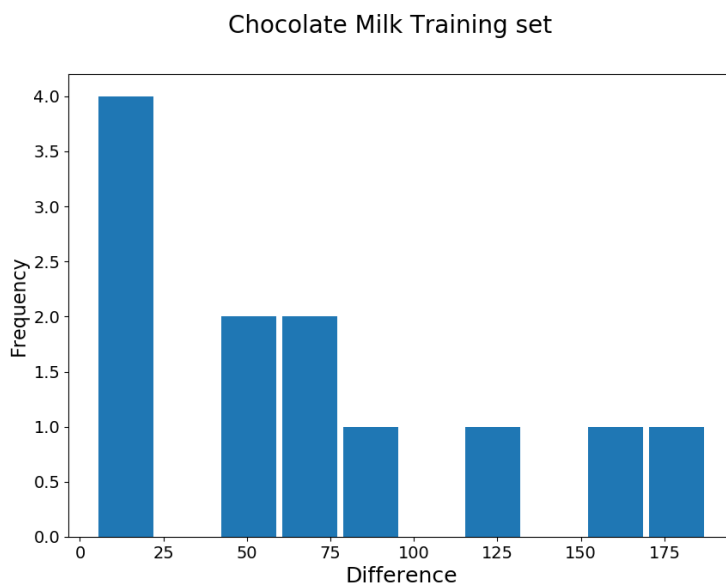


Figure 9: Results for chocolate milk category.

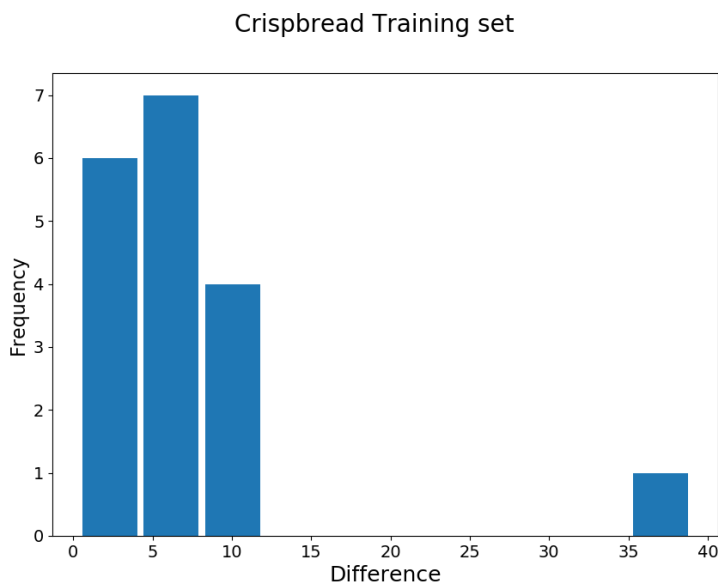


Figure 10: Results for crispbread category.

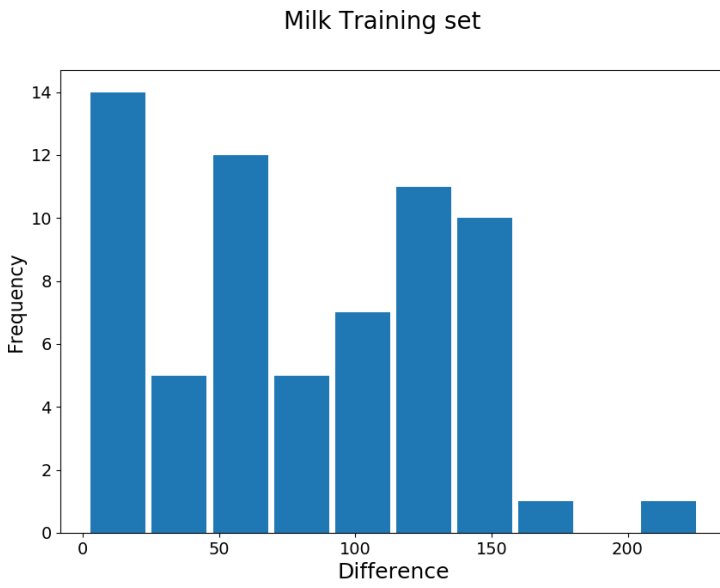


Figure 11: Results for milk category.

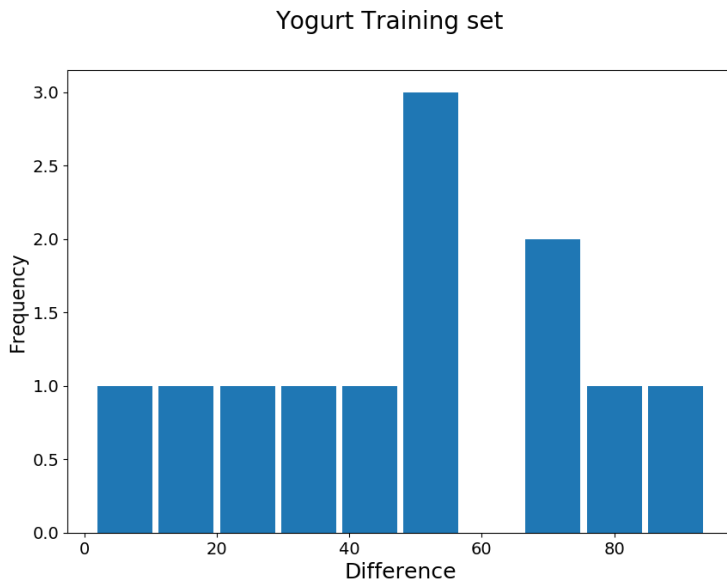


Figure 12: Results for yoghurt category.



UiA University of Agder
Master's thesis
Faculty of Engineering and Science
Department of ICT

© 2019 Runar Isaksen

Eirik Bø Knudsen

Aline Iyagizezeza Walde. All rights reserved