FOOD AND DRINK IMAGE DETECTION AND RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Simon Mezgec

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Barbara Koroušić Seljak, Jožef Stefan International Postgraduate School and Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Prof. Aleš Ude, Chair, Jožef Stefan International Postgraduate School and Jožef Stefan Institute, Ljubljana, Slovenia
Asst. Prof. Peter Rogelj, Member, University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia
Prof. Gregor Papa, Member, Jožef Stefan International Postgraduate School and Jožef Stefan Institute, Ljubljana, Slovenia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Simon Mezgec

FOOD AND DRINK IMAGE DETECTION AND RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Doctoral Dissertation

ZAZNAVANJE IN RAZPOZNAVANJE SLIK HRANE IN PIJAČE Z UPORABO GLOBOKIH KONVOLUCIJSKIH NEVRONSKIH MREŽ

Doktorska disertacija

Supervisor: Prof. Barbara Koroušić Seljak

Ljubljana, Slovenia, October 2021

To everyone who shares life's road with me.

Acknowledgments

The author of this doctoral dissertation would like to thank the dissertation supervisor, Prof. Barbara Koroušić Seljak, for her guidance and invaluable help throughout the entire process of conducting research and writing the dissertation. The author would also like to thank Tamara Bucher from the University of Newcastle, Australia, for providing the fake-food image dataset; Drago Torkar for providing feedback regarding the computer vision aspects of this work; Javier de la Cueva for providing advice regarding web image copyrights; and all co-authors of the papers that make up the main part of the dissertation for their respective contributions to these papers.

The work, presented in this dissertation, was supported by the following projects:

- ERA Chair for Isotope Techniques in Food Quality, Safety and Traceability (ISO-FOOD), which received funding from the European Union's Seventh Framework Program under Grant Agreement Number 621329.
- mHealth Platform for Parkinson's Disease Management (PD_manager), which received funding from the European Union's Horizon 2020 Framework Program under Grant Agreement Number 643706.
- Research Infrastructure on Consumer Health and Food Intake using E-Science with Linked Data Sharing (RICHFIELDS), which received funding from the European Union's Horizon 2020 Framework Program under Grant Agreement Number 654280.
- Supporting Active Ageing through Multimodal Coaching (SAAM), which received funding from the European Union's Horizon 2020 Framework Program under Grant Agreement Number 769661.
- Food Nutrition Security Cloud (FNS-Cloud), which received funding from the European Union's Horizon 2020 Framework Program under Grant Agreement Number 863059.

The research work additionally received financial support from the Slovenian Research Agency (Research Core Funding Number P2-0098). The European Union and the Slovenian Research Agency had no role in the design, analysis or writing of this dissertation.

Abstract

A healthy diet is becoming increasingly relevant as recognizing dietary deficiencies often leads to actionable results that can improve the individual's overall health. However, to identify areas of potential improvement, tracking food intake is necessary. Manual methods have traditionally been used to perform this tracking, but these methods have a number of downsides, such as inaccuracy and a high level of effort and motivation needed to manually track intake. This is why novel solutions are required. Such solutions can efficiently automate food tracking, thus facilitating dietary assessment. Due to the pervasiveness of smartphones with built-in cameras, automating dietary assessment by recognizing food and drink items from images that may not be of the best quality seems like a promising approach to develop solutions that could reach a large portion of the population. There have been multiple approaches presented for this problem, with deep learning—or more specifically, deep neural networks—achieving the state of the art in the field.

This doctoral dissertation presents three solutions for food and drink image detection, recognition, and segmentation using deep convolutional neural networks, which are a type of deep neural networks mainly used for image processing. The first solution includes a detection model to remove nonfood images from a self-acquired dataset, and an image recognition model based on a novel deep neural network architecture, called NutriNet. With it, a classification accuracy of 86.72% was achieved. The second solution is based on fake food (food replicas), which is used in experimental research in behavioral nutrition. Using an existing deep neural network architecture, an image segmentation model was trained on a fake-food image dataset and it achieved an accuracy of 92.18%. The third solution is based on the second one and it was submitted to a worldwide competition for food image recognition, the Food Recognition Challenge. In the scope of this challenge, an image segmentation model was trained and it achieved a precision of 59.2% on the challenging competition dataset of real-world food and drink images, which ranked second in the second round of the competition.

These solutions and results contributed to the development of the food image recognition field in recent years and they further validate the usage of deep convolutional neural networks for this problem, as well as present a novel architecture and approach to input data collection in the deep learning field. To the best of the author's knowledge, they also achieved multiple firsts: the NutriNet solution was the first to recognize images of drinks, while the fake-food solution was the first to automatically recognize food replicas and also the first to include a single deep neural network architecture for the joint segmentation and classification of food images.

Povzetek

Zdrava prehrana postaja čedalje pomembnejša, saj spoznanja glede pomanjkljivosti v prehrani pogosto vodijo do zaključkov, s pomočjo katerih lahko posameznik izboljša svoje zdravstveno stanje. Za prepoznavanje mogočih področij za izboljšave pa je potrebno beležiti vnos hrane. Tradicionalno so bile za to uporabljene ročne metode, vendar so zanje značilne številne pomanjkljivosti, kot sta nenatančnost ter visok nivo zahtevanega truda in motivacije za ročno beleženje vnosa hrane. Zaradi tega so za bolj učinkovito beleženje potrebne nove rešitve, ki lahko avtomatizirajo ta postopek in s tem poenostavijo ocenjevanje prehrane. Zaradi vseprisotnosti pametnih mobilnih telefonov z vgrajenimi fotoaparati je avtomatizacija ocenjevanja prehrane s pomočjo razpoznavanja slik hrane in pijače, ki niso nujno najboljše kakovosti, obetaven pristop za razvoj rešitev, ki lahko dosežejo visok delež prebivalstva. Za reševanje tega problema je bilo predstavljenih več pristopov, najboljše rezultate pa so dosegle rešitve, ki uporabljajo globoko učenje oziroma natančneje—globoke nevronske mreže.

V tej doktorski disertaciji so predstavljene tri rešitve za zaznavanje, razpoznavanje in segmentacijo slik hrane in pijače z uporabo globokih konvolucijskih nevronskih mrež. Te so vrsta globokih nevronskih mrež, ki se večinoma uporabljajo za obdelovanje slik. Prva rešitev vključuje model za zaznavanje slik, ki ne vsebujejo hrane, v samostojno pridobljeni podatkovni zbirki slik ter model za razpoznavanje, ki temelji na novi arhitekturi globokih nevronskih mrež, imenovani NutriNet. S to arhitekturo je bila dosežena 86,72 % klasifikacijska točnost. Druga rešitev temelji na lažni hrani (replike hrane), ki se uporablja v eksperimentalnih raziskavah prehranjevalnega vedenja. Z uporabo obstoječe arhitekture globokih nevronskih mrež je bil naučen model za segmentacijo na podatkovni zbirki slik lažne hrane, ki je dosegel 92,18 % točnost. Tretja rešitev temelji na drugi rešitvi in oddana je bila v sklopu svetovnega tekmovanja v razpoznavanju slik hrane Food Recognition Challenge. V okviru tekmovanja je bil naučen model za segmentacijo, ki je dosegel 59,2 % natančnost na zahtevni podatkovni zbirki tekmovanja, ki vsebuje slike hrane in pijače iz resničnega sveta, s čimer je ta rešitev dosegla drugo mesto v drugi rundi tekmovanja.

Predstavljene rešitve in rezultati so v preteklih letih prispevali k razvoju področja razpoznavanja slik hrane in dodatno potrjujejo smiselnost uporabe globokih konvolucijskih nevronskih mrež za reševanje tega problema. Na področju globokega učenja je prispevek predstavljenih rešitev nova arhitektura ter pristop k zbiranju vhodnih podatkov. Kolikor je avtorju znano, so bile te rešitve tudi v več pogledih prve: rešitev, ki temelji na arhitekturi NutriNet, je bila prva, ki je razpoznavala slike pijač, medtem ko je bila rešitev, ki temelji na lažni hrani, prva, ki je samodejno razpoznavala replike hrane, in tudi prva, ki je vsebovala enotno arhitekturo globokih konvolucijskih nevronskih mrež za skupno segmentacijo in klasifikacijo slik hrane.

Contents

List of Figures

Abbreviations

DCNN	...	deep convolutional neural network
FCN	...	fully convolutional network
FFB	...	fake food buffet
FRC	...	Food Recognition Challenge
HTC	...	hybrid task cascade
IF	...	impact factor
ResNet	...	residual neural network
UNIMIB2016	...	University of Milano-Bicocca 2016

Chapter 1

Introduction

1.1 Motivation

In recent years, more and more people are becoming aware of the importance of a healthy lifestyle, a large part of which is a healthy diet. As individuals are trying to improve the quality of their diet, it is necessary to first identify aspects of their diet that can be improved. This can be done by analyzing and evaluating their current food intake, which is referred to as dietary assessment.

In the past, dietary assessment was performed using manual methods. These methods often require self-reporting by the individual and include approaches like nutrition questionnaires, 24-hour dietary recall, and others. Because they are performed manually, and because they can require action by the individual, they are time-consuming and prone to errors, as, for example, food quantity can be very challenging to assess accurately, and individuals can lose motivation to constantly track every food or drink they ingest. As a consequence of these challenges, dietary assessment was mostly utilized only by people who had to perform it, such as by patients with conditions that necessitate a close analysis of food intake.

A simpler and more straightforward approach with less potential for errors is thus needed to improve this process. This is why research into automated food recognition has recently become a popular area of research. There are multiple approaches to this automation, with one of the most promising ones being food image recognition. This particular approach aims to automatically provide nutritional values for different foods by taking an image of a food or drink item and matching its name with food composition databases. Since smartphones with built-in cameras are so prevalent, the barrier to entry for using such solutions is very low. Consequently, this approach has the potential to enable the development of applications that reach a large part of the general population.

1.2 Approach

The goal of food image recognition is to recognize food and drink items that are present in an image. This is done by first assigning features to the items and then classifying them into the appropriate food classes. Traditionally, this was performed using manually-defined feature descriptors. There has been a large variety of manual methods presented for food image recognition—these include multiple kernel learning [1], bag-of-features [2], and pairwise local features [3], among others. The issue with manually-defined methods is that they mostly achieve a low classification accuracy. This is due to the nature of food items, which makes their recognition a very challenging computer vision problem.

There are multiple reasons why it is difficult to accurately recognize food and drink items [4]. Food items are often deformable, which means their appearance can vary significantly from image to image. On the other hand, different food items can have a very similar appearance, making them difficult to distinguish from one another. Additionally, a large amount of visual information is lost in the preparation of a dish, and drink items specifically contain little visual information that is useful for recognition—often only color, quantity and container information. Finally, there is a very large number of different foods and drinks, and even the same dish can have many different variations based on the characteristics of the local cuisine.

The consequence of these challenges is that the state of the art in the food image recognition research field was achieved by a method that automates the process of feature definition. By doing that, it can automatically learn what features are optimal for the differentiation of food and drink items. This method is deep learning, or deep neural networks. These networks were inspired by the way in which the biological brains work, and they are composed of layers of neurons. As the input image progresses through these layers, the network learns more and more complex features—from simple features, such as edges and shapes, to complex features, like the type of food. Deep neural networks allow the training of models that automatically learn features from a set of input images. Due to this, they are capable of recognizing complex objects, such as food, with a very high accuracy, which is why they achieved the state of the art for multiple computer vision problems [5].

Specifically, deep convolutional neural networks (DCNNs) are the most widely used networks for the task of food image recognition. These networks mimic the visual model of animals, thus trying to gain an understanding of the image in a similar way that animals do [6]. There are multiple types of layers typically found in DCNNs, with the most distinctive being the convolutional layer, which contains learnable filters that aim to learn local features of an image. DCNNs have been able to achieve promising results for food image recognition [7]–[9]. In particular, they have been used in solutions that detect food images [10], recognize food items [11], and segment food images [12], which refers to the process of partitioning the food image into separate food classes, along with the background. Since the size and diversity of input datasets are crucial to the accuracy of trained DCNN models, multiple food image datasets have been made publicly available [12]–[15], which accelerated progress in the field.

However, most prior research in the field was still limited in its real-world applicability, either due to the low classification accuracy or the limitations of the image datasets used. Namely, the majority of solutions either used datasets, gathered in controlled environments, or datasets that contain a low number of different food classes, or both, and to the best of the author's knowledge, there were no prior solutions that would be able to recognize images of drinks. All of these issues mean that such solutions are not robust enough to be deployed in real-world applications. This was the motivation behind the research, presented in this dissertation. Each additional food class also means added complexity, generally decreasing the overall classification accuracy of trained DCNN models. The goal was thus also to research the possibility of developing a more accurate DCNN architecture for food and drink image recognition.

1.3 Research Hypotheses

Two hypotheses were defined in the scope of this doctoral dissertation. These hypotheses are listed below and further elaborated on in Chapter 5.

1.3.1 First Hypothesis

To improve the classification accuracy of food and drink image recognition, a new deep convolutional neural network architecture can be designed.

1.3.2 Second Hypothesis

A single deep convolutional neural network architecture can be used to automatically and jointly perform the segmentation and classification of food and drink images.

1.4 Scientific Methods and Contributions

This doctoral dissertation includes three publications as the main part of the dissertation. These publications are articles, which were published in international peer-reviewed impact factor (IF) journals, and are as follows:

1. *S. Mezgec and B. K. Seljak, "NutriNet: A deep learning food and drink image recognition system for dietary assessment,"* Nutrients*, vol. 9, no. 7, p. 657, 2017* [4]. IF: 5.717, 111 citations (25 October 2021) [16].

2. *S. Mezgec, T. Eftimov, T. Bucher, and B. K. Seljak, "Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment,"* Public Health Nutrition*, vol. 22, no. 7, pp. 1193–1202, 2019* [17]. IF: 4.022, 24 citations (25 October 2021) [16].

3. *S. Mezgec and B. K. Seljak, "Deep neural networks for image-based dietary assessment,"* Journal of Visualized Experiments*, vol. 169, e61906, 2021* [18]. IF: 1.4, 0 citations (25 October 2021) [16].

To address the first hypothesis, multiple popular DCNN architectures were tested on a self-acquired dataset of food and drink images. To improve the classification accuracy, one of these architectures was taken as the basis for the development of a novel DCNN architecture, called NutriNet. This architecture was then tested against the popular DCNN architectures on the same dataset, as well as on a dataset of real-world images, and on a publicly available food image dataset. The results are presented and analyzed in the first publication [4].

For the second hypothesis, a dataset of fake-food images was first sourced. Fake food (food replicas) is used to perform behavioral studies. To facilitate the analysis of the results from these studies, a fake-food image recognition approach was developed. This included using an existing DCNN architecture to train a model on the fake-food image dataset, which segments the image into individual food and drink items and then classifies them. The training and testing results, as well as the implications of automating the recognition of food replicas in behavioral studies, are presented in the second publication [17]. In order to perform food and drink image segmentation and classification on real food, another approach was developed in the scope of the Food Recognition Challenge (FRC) [19], which is an international competition for food image recognition. This approach is presented in the third publication [18], and it was also used to implement a mobile application for dietary assessment, which is described in Chapter 5.

The research work, as well as the writing of the articles themselves, for both the first [4] and third publication [18], was performed by the author of this doctoral dissertation. The dissertation author is therefore the first author of these publications, with the second author being the dissertation supervisor, Prof. Barbara Koroušić Seljak, who oversaw the

research and edited the article. As for the second publication [17]—the research work was split into two parts: fake-food image recognition and linking the recognition results to food composition databases. The first part was performed by the dissertation author, who is also the first author of the publication, whereas the second part was performed by the second author, Tome Eftimov. The third author, Tamara Bucher, provided the fake-food image dataset, whereas the dissertation supervisor oversaw the research work. All authors of that publication contributed to the writing and editing of the article.

In addition to the three publications that make up the main part of the dissertation, the research work, presented in this dissertation, resulted in another article that was published in an IF journal:

- *N. V. Matusheski, A. Caffrey, L. Christensen, S. Mezgec, S. Surendran, M. F. Hjorth, H. McNulty, K. Pentieva, H. M. Roager, B. K. Seljak, K. S. Vimaleswaran, M. Remmers, and S. Péter, "Diets, nutrients, genes and the microbiome: Recent advances in personalised nutrition,"* British Journal of Nutrition*, pp. 1–9, 2021* [20]. IF: 3.718, 3 citations (25 October 2021) [16].

The dissertation author's research work resulted in the finalist selection for the 2019 DSM Bright Science Award [21], along with a corresponding presentation at the 13th European Nutrition Conference [22], and this publication summarizes the work by all four finalists. Since the publication describes solutions that are already presented in the dissertation publications, and because the dissertation author's contribution was only one of the four main parts of that article, it is not included in this dissertation. Finally, the research work from this dissertation was also described in a conference paper [23].

1.5 Dissertation Structure

This doctoral dissertation is structured as follows: Chapter 2 contains the first of the publications [4], listed in Section 1.4, Chapter 3 contains the second publication [17], and Chapter 4 contains the third publication [18]. All three publications are included exactly as they appear in their respective journals, each with its own page and section numbering, abstract, and references. This is done to facilitate navigation through this dissertation. As such, the List of Figures, Abbreviations, and References of this dissertation apply only to the dissertation text itself, and not to the publications. Finally, Chapter 5 contains the discussion portion of the dissertation, and Chapter 6 summarizes the dissertation and gives final remarks.

Chapter 2

NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment

The first goal of the conducted research work was to build upon prior results and findings in the field and address their limitations, which were mainly a low classification accuracy and a low number of recognizable food items, and consequently develop a mobile application for dietary assessment by using an accurate and efficient DCNN model for food image recognition as part of it. To train such a model, an extensive food image dataset was needed. However, there were no suitable food image datasets that would satisfy the necessary conditions of including a sufficient number of different food items, including any drink items at all, and containing enough images for each individual food item.

A new dataset thus needed to be built, which was done by downloading food and drink images from a list of 520 items using results from web searches. Since these queries usually result in irrelevant images as well, this dataset needed to be cleaned. This was done by acquiring a different dataset—one that contained food and drink images in one class, and images of other objects in the second class. Using this dataset, a food/nonfood DCNN model was trained. By running this food detection model on the first dataset, erroneous results were removed from it. After that, data augmentation was performed on the dataset to generate new image variations.

NutriNet, a novel DCNN architecture, was developed by modifying the popular AlexNet architecture [24]. This was done with the goal of improving its classification accuracy, as well as its training efficiency. NutriNet was tested on the aforementioned dataset against AlexNet, GoogLeNet [25], and residual neural networks (ResNet) [26]. Additionally, the performance of these architectures was tested on a small real-world food image dataset, as well as on the publicly available University of Milano-Bicocca 2016 (UNIMIB2016) food image dataset [12]. All results are included and discussed in the publication below.

Permission to include the publication *"NutriNet: A deep learning food and drink image recognition system for dietary assessment"* [4] in this doctoral dissertation was confirmed by the journal *Nutrients* in an email exchange from 7 June 2021.

nutrients

MDPI

Article

NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment

Simon Mezgec [1,*] and Barbara Koroušić Seljak [2]

[1] Information and Communication Technologies, Jožef Stefan International Postgraduate School, Jamova Cesta 39, 1000 Ljubljana, Slovenia
[2] Computer Systems Department, Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia; barbara.korousic@ijs.si
* Correspondence: simon.mezgec@gmail.com; Tel.: +386-51-686-385

Received: 10 March 2017; Accepted: 20 June 2017; Published: 27 June 2017

Abstract: Automatic food image recognition systems are alleviating the process of food-intake estimation and dietary assessment. However, due to the nature of food images, their recognition is a particularly challenging task, which is why traditional approaches in the field have achieved a low classification accuracy. Deep neural networks have outperformed such solutions, and we present a novel approach to the problem of food and drink image detection and recognition that uses a newly-defined deep convolutional neural network architecture, called NutriNet. This architecture was tuned on a recognition dataset containing 225,953 512 × 512 pixel images of 520 different food and drink items from a broad spectrum of food groups, on which we achieved a classification accuracy of 86.72%, along with an accuracy of 94.47% on a detection dataset containing 130,517 images. We also performed a real-world test on a dataset of self-acquired images, combined with images from Parkinson's disease patients, all taken using a smartphone camera, achieving a top-five accuracy of 55%, which is an encouraging result for real-world images. Additionally, we tested NutriNet on the University of Milano-Bicocca 2016 (UNIMIB2016) food image dataset, on which we improved upon the provided baseline recognition result. An online training component was implemented to continually fine-tune the food and drink recognition model on new images. The model is being used in practice as part of a mobile app for the dietary assessment of Parkinson's disease patients.

Keywords: NutriNet; deep convolutional neural networks; deep learning; food recognition; food detection; drink recognition; drink detection; Parkinson's disease

1. Introduction

As people are becoming increasingly aware of the importance of a healthy diet, the need for automatic food and drink recognition systems has arisen. Not only can such systems provide the automatic recognition of food and drink items, but they can also enable an estimation of their nutritional values, making them especially useful for dietary assessment and planning, which is applicable for patients with different dietary restrictions, as well as for healthy individuals by preventing nutrition-related conditions.

The problem of food and drink image detection and recognition is challenging due to the nature of food and drink items. Foods are typically deformable objects, which makes the process of defining their structure difficult. Furthermore, some food types can have a high intra-class (similar foods look very different) and low inter-class (different foods look very similar) variance, making the process of specifying the food type even more challenging. The issue with drink recognition is that there is only a limited amount of information that can be gained using images of drink items; an example of such information is the drink's color, whether the drink is well-lit, and the drink's density. All of these

obstacles make food and drink image detection and recognition a particularly challenging computer vision problem.

We approached this problem using deep learning or deep neural networks [1]. Many problems in computer vision require the definition of complex features that are very challenging and time consuming to manually define. Deep learning alleviates this as it allows computational models composed of multiple processing layers to automatically learn these features and represent the input data with them. Deep learning models have substantially improved the best results in a variety of research fields, computer vision being one of them [1]. Specifically, we are using deep convolutional neural networks, which are a type of deep neural network that is inspired by the visual cortex of animals, where the individual neurons react to overlapping regions in the visual field [2]. This makes convolutional neural networks especially suitable for computer vision, as the goal of computer vision systems is the same as that of animal vision systems: to gain an understanding of input images.

When an image is fed through a convolutional neural network, a series of operations is performed on the image as data transitions through the network layers. The layer parameters are then adjusted in each iteration, which is how the training is performed. Three types of layers are the most common in convolutional neural networks: convolutional, fully-connected and pooling layers. Convolutional layers contain learnable filters that are trained in such a way that they respond to certain features in the input data; an example of learned filters is shown in Figure 1. Fully-connected layers, on the other hand, compose output data from other layers to gain higher-level knowledge from it. The pooling layers down-sample the input data, but since this layer type does not accept parameters, it is usually not counted towards the total neural network layer depth.

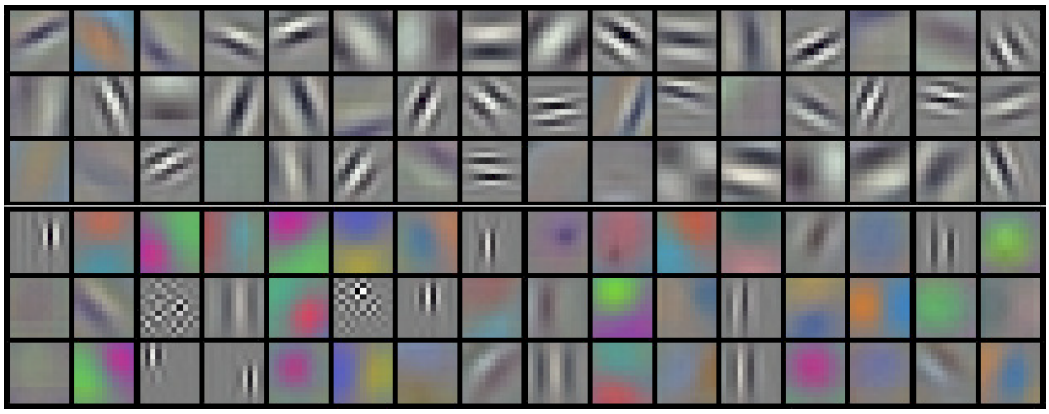

Figure 1. Example filters by Krizhevsky et al. [3]. Because these filters were learned using the first convolutional layer of the neural network, the represented features are simple, such as the edge orientation and frequency (learned features become progressively more complex with each additional convolutional layer). Reproduced with permission from Alex Krizhevsky, Advances in NIPS 25; published by Curran Associates, Inc., 2012.

The structure of this paper is as follows. In Section 1.1, related work in the field of food image detection and recognition is presented; in Section 2.1, our image datasets and their acquisition are described; Section 2.2 contains information about NutriNet and other convolutional neural network models that were tested, along with their training process; Section 2.3 describes how the online training component was implemented; in Section 3, the training and testing results of the deep neural network models are given; in Section 3.1, we present testing results of the NutriNet architecture on other datasets, including a real-world image dataset that we built for this purpose; Section 4 contains the discussion part of the research work; and Section 5 concludes the paper and gives an overview of the work done, as well as possible future work.

1.1. Related Work

While there have not been any dedicated drink image recognition systems, there have been multiple approaches to food image recognition in the past, and we will briefly mention the most important ones here. In 2009, an extensive food image and video dataset was built to encourage further research in the field: the Pittsburgh Fast-Food Image Dataset (PFID), containing 4545 still images, 606 stereo image pairs, 303 360° food videos and 27 eating videos of 101 different food items, such as "chicken nuggets" and "cheese pizza" [4]. Unfortunately, this dataset focuses only on fast-food items, not on foods in general. The authors provided the results of two baseline recognition methods tested on the PFID dataset, both using an SVM (Support Vector Machine) classifier to differentiate between the learned features; they achieved a classification accuracy of 11% with the color histogram method and 24% with the bag-of-SIFT-features method. The latter method counts the occurrences of local image features described by the popular SIFT (Scale-Invariant Feature Transform) descriptor [5]. These two methods were chosen based on their popularity in computer vision applications, but the low classification accuracy showed that food image recognition is a challenging computer vision task, requiring a more complex feature representation.

In the same year, a food image recognition system that uses the multiple kernel learning method was introduced, which tested different feature extractors, and their combination, on a self-acquired dataset [6]. This proved to be a step in the right direction, as the authors achieved an accuracy of 26% to 38% for the individual features they used and an accuracy of 61.34% when these features were combined; the features include color, texture and SIFT information. Upon conducting a real-world test on 166 food images taken with mobile phones, the authors reported a lower classification accuracy of 37.35%, which was due to factors like occlusion, noise and additional items being present in the real-world images. The fact that the combination of features performed better than the individual features further hinted at the need for a more in-depth representation of the food images. Next year, the pairwise local features method, which applies the specifics of food images to their recognition, was presented [7]. This method analyzes the ingredient relations in the food image, such as the relations between bread and meat in a sandwich, by computing pairwise statistics between the local features. The authors performed an evaluation of their algorithm on the PFID dataset and achieved an accuracy of 19% to 28%, depending on which measure they employed in the pairwise local features method. However, they also noted that the dataset had narrowly-defined food classes, and after joining them into 7 classes, they reported an accuracy of 69% to 78%. This further confirmed the limitations of food image recognition approaches of that time: if a food image recognition algorithm achieved a high classification accuracy, it was only because the food classes were very general (e.g., "chicken").

In 2014, another approach was presented that uses an optimized bag-of-features model for food image recognition [8]. The authors tested 14 different color and texture descriptors for this model and found that the HSV-SIFT descriptor provided the best result. This descriptor describes the local textures in all three color channels of the HSV color space. The model was tested on a food image dataset that was built for the aims of the project Type 1 Diabetes Self-Management and Carbohydrate Counting: A Computer Vision Based Approach (GoCARB) [9], in the scope of which they constructed a food recognition system for diabetes patients. The authors achieved an accuracy of 77.80%, which was considerably higher than previous approaches.

All of the previously-described solutions are based on manually-defined feature extractors that rely on specific features, such as color or texture, to recognize the entire range of food images. Furthermore, the images used in the recognition systems presented in these solutions were taken under strict conditions, containing only one food dish per image and often perfectly cropped. The images that contained multiple items were manually segmented and annotated, so the final inputs for these hand-crafted recognition systems were always ideally-prepared images. The results from these research works are therefore not indicative of general real-world performance due to the same problems with real-world images as listed above.

These issues show that hand-crafted approaches are not ideal for a task as complex as food image recognition, where it seems the best approach is to use a complex combination of a large number of features, which is why deep convolutional neural networks, a method that automatically learns appropriate image features, achieved the best results in the field. Deep neural networks can also learn to disregard surrounding noise with sufficient training data, eliminating the need for perfect image cropping. Another approach for the image segmentation task is to train a neural network that performs semantic segmentation, which directly assigns class labels to each region of the input image [10,11]. Furthermore, deep neural networks can be trained in such a way that they perform both object detection and recognition in the same network [12,13].

In 2014, Kawano et al. used deep convolutional neural networks to complement hand-crafted image features [14] and achieved a 72.26% accuracy on the University of Electro-Communications Food 100 (UEC-FOOD100) dataset that was made publicly available in 2012 [15]; this was the highest accuracy on the dataset at that time. Also in 2014, a larger version of the UEC-FOOD100 dataset was introduced, the University of Electro-Communications Food 256 (UEC-FOOD256), which contains 256 as opposed to 100 food classes [16]; while UEC-FOOD100 is composed of mostly Japanese food dishes, UEC-FOOD256 expands on this dataset with some international dishes. At that time, another food image dataset was made publicly available: the Food-101 dataset. This dataset contains 101,000 images of 101 different food items, and the authors used the popular random forest method for the recognition task, with which they achieved an accuracy of 50.76% [17]. They reported that while this result outperformed other hand-crafted efforts, it could not match the accuracy that deep learning approaches provided. This was further confirmed by the subsequently published research works, such as by Kagaya et al., who tested both food detection and food recognition using deep convolutional neural networks on a self-acquired dataset and achieved encouraging results: a classification accuracy of 73.70% for the recognition and 93.80% for the detection task [18]. In 2015, Yanai et al. improved on the best UEC-FOOD100 result, again with deep convolutional neural networks, only this time, with pre-training on the ImageNet dataset [19]. The accuracy they achieved was 78.77% [20]. A few months later, Christodoulidis et al. presented their own food recognition system that uses deep convolutional neural networks, and with it, they achieved an accuracy of 84.90% on a self-acquired and manually-annotated dataset [21].

In 2016, Singla et al. used the famous GoogLeNet deep learning architecture [22], which is described in Section 2.2, on two datasets of food images, collected using cameras and combined with images from existing image datasets and social media. With a pre-trained model, they reported a recognition accuracy of 83.60% and a detection accuracy of 99.20% [23]. Also in 2016, Liu et al. achieved similarly encouraging results on the UEC-FOOD100, UEC-FOOD256 and Food-101 datasets by using an optimized convolution technique in their neural network architecture [24], which allowed them to reach an accuracy of 76.30%, 54.70% and 77.40%, respectively. Furthermore, Tanno et al. introduced DeepFoodCam, which is a smartphone food image recognition application that uses deep convolutional neural networks with a focus on recognition speed [25]. Another food image dataset was made publicly available in that year: the University of Milano-Bicocca 2016 (UNIMIB2016) dataset [26]. This dataset is composed of images of 1027 food trays from an Italian canteen, containing a total of 3616 food instances, divided into 73 food classes. The authors tested a combined segmentation and recognition deep convolutional neural network model on this dataset and achieved an accuracy of 78.30%. Finally, in 2016, Hassannejad et al. achieved the current best classification accuracy values of 81.45% on the UEC-FOOD100 dataset, 76.17% on the UEC-FOOD256 dataset and 88.28% on the Food-101 dataset [27]. All three results were obtained by using a deep neural network model based on the Google architecture Inception; this architecture is the basis for the previously-mentioned GoogLeNet.

It seems that deep learning is a very promising approach in the field of food image recognition. Previous deep learning research reported high classification accuracy values, thus confirming the viability of the approach, but they focused on smaller food image datasets, often limited to 100 different food items or less. Moreover, none of these solutions recognize drink images. In this paper, we will

present our solution that addresses these issues. We developed a new deep convolutional neural network architecture called NutriNet and trained it on images acquired from web searches for individual food and drink items. With this architecture, we achieved a higher classification accuracy than most of the results presented above and found that, on our recognition dataset, it performs better than AlexNet, which is the deep learning architecture it is based on; the results are described in-depth in Section 3. Additionally, we developed an online training component that automatically fine-tunes the deep learning image recognition model upon receiving new images from users, thus increasing the number of recognizable items and the classification accuracy over time. The online training is described in Section 2.3.

By trying to solve the computer vision problem of recognizing food and drink items from images, we are hoping to alleviate the issue of dietary assessment, which is why our recognition system is integrated into the PD Nutrition application for the dietary assessment of Parkinson's disease patients [28], which is being developed in the scope of the project mHealth Platform for Parkinson's Disease Management (PD_manager) [29]. In practice, the system works in the following way: Parkinson's disease patients take an image of food or drink items using a smartphone camera, and our system performs recognition using deep convolutional neural networks on this image. The result is a food or drink label, which is then matched against a database of nutritional information, thus providing the patients with an automatic solution for food logging and dietary assessment.

2. Materials and Methods

2.1. Food and Drink Image Datasets

An extensive image dataset is critical for a food and drink image recognition system because it enables the learning of more general features and therefore helps combat overfitting, which is a common occurrence in machine learning, where a model describes random noise instead of learning generalizable knowledge. The goal was therefore to build a dataset that contains as many food and drink items as possible and where each item is represented with as many images as possible. Additionally, we also wanted to have images of foods and drinks that are local to the Central European region, since that would yield better results in the final application of the dietary-assessment system. This is because food and drink types vary by region, meaning that a localized image dataset offers a more accurate representation of the foods and drinks that would be recognized in practice. However, it is important to note that the entire data-preparation and model-training process, as well as the online training component are not specific to images of Central European foods and drinks; images and class labels could be provided for other foods and drinks and in other languages.

We first tried building the image recognition dataset using publicly-available images from recipe-gathering websites. This seemed appropriate since popular recipe websites have a large number of users, most of whom post not only the recipes themselves, but also images of the final product. However, this approach had two crucial drawbacks: First, the only useful labels the recipes contained were food categories (e.g., "meat dishes", "vegetable dishes", etc.), rather than specific dishes or drinks. The web pages for specific recipes also contained the recipe name, but since there were no naming rules or pre-defined dishes, the names were sometimes different for the same dish and very similar for different types of dishes. This meant that the recognition result would have to be a very general class, which contained very different food and drink items, making the recognition difficult. Additionally, since the results would be so general, the usefulness of such a model is questionable. Second, the resulting image dataset was too small to train a high-quality model: it contained less than 10,000 images because all of the images were taken from one recipe-gathering website. The reason why only one website was used is that food classes vary substantially from website to website, and a unified recipe image dataset was therefore impossible.

That is why we changed our approach and built the image dataset in a different way. Using existing food and drink class labels from the PD Nutrition dietary-assessment system, a web image search was

performed for each food and drink class, and the resulting images were saved locally. To achieve this, we used the Google Custom Search API [30] inside a Python (Python Version 2.7.6 was used – Python is developed by the Python Software Foundation, Beaverton, OR, USA [31]) script that reads the class text labels and performs a Google image search for every label. Each class represents a food or drink item, and the script creates a folder for every class and stores its top image results. We chose to save 100 images per item; this offered a suitable balance between image quality and quantity, as too few images meant the dataset was not sufficiently large, and too many images meant saving a large number of images that do not necessarily contain the searched food or drink item, since Google image search returns the best results first. All of the images that we acquired are freely downloadable online and are labeled as either "Creative Commons Public Domain" or "Creative Commons No Derivatives".

As a result, this recognition dataset we built had 520 food and drink image classes of 100 images each. However, due to the nature of web image searches, some results included irrelevant and low-quality images, as well as duplicate images. This meant that, in order to improve the overall dataset quality, images like that needed to be removed. This was done using a deep convolutional neural network model for image detection, by which we are referring to the process of classifying an image as either a food or drink image or as an image that contains anything else, similar to Kagaya et al. [18]. The detection model is described in Section 2.2.

To train a model like that, a secondary image dataset needed to be built, one that contains food and drink images in one class and images of everything else in the other, which is similar to how Singla et al. structured their detection dataset [23]. This was done by merging the previously-acquired recipe image dataset, which includes images of foods, and also some drinks, and the ImageNet dataset [19]. Using another Python script, images, labeled as food or drink items, were downloaded from the ImageNet dataset, as well as a random subset of all of the other images in the dataset. The entire ImageNet dataset was not saved due to its size, which would significantly increase the training time for the food and drink image detection deep learning model and the dataset would be very unbalanced since there are many more images of other objects than there are of foods and drinks. Additionally, to further reduce the dataset imbalance and gain some rotational invariance, all of the food and drink images were rotated by 90°, 180° and 270°. These four variants per image were then saved: the resulting dataset contains 130,517 images, of which 54,564 images are food/drink images and 75,953 images contain other objects. This detection dataset is depicted in Figure 2.

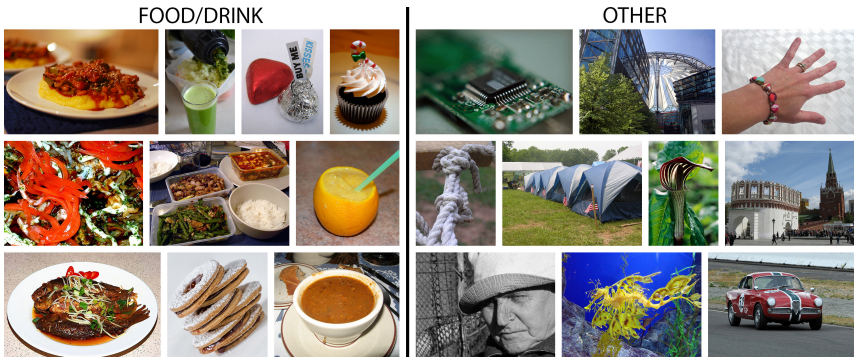

Figure 2. Example images from the two classes of the food and drink image detection dataset, obtained by merging recipe website images and a subset of the ImageNet dataset.

The food and drink image detection model was used on the recognition dataset, and images that were labeled with "other" were removed. As the number of such images varies from class to class, the classes became unbalanced as a consequence. Like with the detection dataset, the remaining images were then rotated for a total of four variations per image, and three additional data-augmentation steps were performed on the recognition dataset to increase the dataset size and gain further invariance;

the images were flipped horizontally; random color noise was introduced to them; and they were zoomed in so that 25% of all of the image borders were removed for each image. In total, there are therefore 7 such variations per image.

The dataset was then divided into training, validation and testing subsets with a 70%/10%/20% split. Additionally, two versions of the recognition dataset were created; they differ only in image size, as the images were resized to 256 × 256 pixels for the first version and to 512 × 512 pixels for the second version. The reason for having two versions of the same dataset is because our NutriNet architecture, along with the other modified architectures we tested, accepts 512 × 512 pixel images, whereas the pre-trained models accept 256 × 256 pixel images; these models and the reasoning behind using different-resolution images are described in Section 2.2. The detection dataset also contains 512 × 512 pixel images, since NutriNet was used as the final detection model. Finally, all of the datasets were transformed into the Lightning Memory-Mapped Database (LMDB) format [32] to enable a higher throughput of input images through the deep learning framework that we used, which is also described in Section 2.2.

Both versions of the recognition dataset contain 225,953 images of 520 different food and drink items; example images from this dataset can be seen in Figure 3. The total size of the transformed LMDB recognition dataset with larger images is 72 GB, whereas the size of the one with smaller images is 23 GB. The size of the transformed detection dataset is 46 GB. The tools needed to download images from the recognition dataset can be downloaded from the Jožef Stefan Institute website [33], and they are also available and described in the Supplementary Materials; this includes the Python script mentioned above, a complete list of all of the food and drink labels we used to create the recognition dataset and a text file with instructions on how to use the script to download the images.

Figure 3. Example images from the final food and drink image recognition dataset, built from Google image searches. Each one of these images represents a different food or drink class.

2.2. NutriNet and Other Deep Convolutional Neural Networks

After the image datasets were acquired, we developed a food and drink image detection and recognition system that uses deep convolutional neural networks. A food or drink image is provided to the recognition model as the input, and the output is a text class label describing the food or drink item. The neural network classifies the input image into one class; if there are more food or drink items present in the image, the most prevalent one is provided as the output. For the detection model, the output is one of the two class labels: "food/drink" or "other".

We used four different deep convolutional neural network architectures: NutriNet, which is the architecture developed in the scope of this research work, and three ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners – AlexNet (2012) [3], GoogLeNet (2014) [22] and Deep Residual Networks (ResNet, 2015) [34]. This annual challenge is one of the most important image recognition challenges, and its winners provide state-of-the-art approaches in the field. Since the challenge tasks competitors with correctly-classifying images into 1000 classes, these three architectures provided a suitable choice for training on food and drink images, as well as a comparison to the NutriNet architecture.

AlexNet is the shallowest of the three pre-existing deep neural network architectures, having five convolutional layers and three fully-connected layers. Being shallower, AlexNet learns less in-depth features, but provides faster learning times. GoogLeNet is somewhat deeper than AlexNet, having a total of 22 layers. For the purpose of this research work, we used the ResNet-152 variant of the ResNet architecture, which has 152 layers and is therefore considerably deeper than the other two ILSVRC-winning architectures. Despite this difference in layer depth, AlexNet accepts roughly the same number of parameters as ResNet, approximately 60 million, whereas GoogLeNet only accepts around four million parameters. This is due to the fact that, unlike AlexNet, GoogLeNet and ResNet do not use fully-connected layers; since these layers contribute the largest proportion of parameters, these two architectures are able to have a much higher number of layers without a considerable increase in the number of parameters. All three of them use dropout, which is a technique to prevent overfitting in neural network models by randomly excluding units in the neural network, along with their connections, during the training process [35].

Our convolutional neural network architecture, NutriNet, is a modification of the AlexNet architecture. The first difference is that while AlexNet, GoogLeNet and ResNet accept 256×256 pixel images and take a 227×227 pixel image patch (224×224 pixel patch for GoogLeNet and ResNet) for processing before the first layer, NutriNet accepts 512×512 pixel images and takes a 454×454 pixel image patch for processing. The reason for the difference in the input image size is that we used pre-trained models for the other deep neural network architectures; the AlexNet, GoogLeNet and ResNet models were all pre-trained on the previously-described ImageNet dataset, and these models accept 256×256 pixel images. On the other hand, we wanted to extract as much information from our dataset images as possible, which is why we used higher resolution images for the NutriNet architecture. Additionally, we modified the other architectures so that they accept 512×512 pixel images and included models using these modified architectures in the training and testing process. This was done to gain an understanding of whether a potential difference in classification accuracy between the ILSVRC-winning architectures and NutriNet is due to the higher-resolution images or due to the NutriNet architecture. All of the results are presented in Section 3. The main reason an image patch is randomly cropped before the first neural network layer is to gain some translational invariance [36].

The second difference is that NutriNet has an additional convolutional layer at the beginning of the neural network compared to AlexNet, which means it has 6 convolutional layers in total. This convolutional layer was added to gain additional knowledge about the features in the higher resolution images. To test whether adding any further convolutional layers to the architecture would yield better results, we also tested NutriNet with an extra convolutional layer added after the input layer; we are calling this architecture NutriNet+. Lastly, as a consequence of the different input image resolutions and the additional convolutional layer, the dimensionality of the layer outputs is different. Due to this difference in dimensionality at the first fully-connected layer, NutriNet contains a considerably lower number of parameters than AlexNet: approximately 33 million. Figure 4 contains a diagram of the image classification process using the NutriNet architecture on an example image from the recognition dataset.

Using the aforementioned architectures, multiple network training parameters were defined and tested: solver type, learning rate, number of epochs and batch size. The solver type determines the method that minimizes the loss function, which is the primary quality measure in the neural network

training process. The learning rate defines the rate with which the neural network's parameters are being changed during training: higher learning rates speed up the training process, but can converge to worse loss values than lower rates. The number of epochs is the number of times all of the training images are fed through the neural network, while the batch size determines how many images are fed through at the same time. The results of this testing are presented in Section 3.

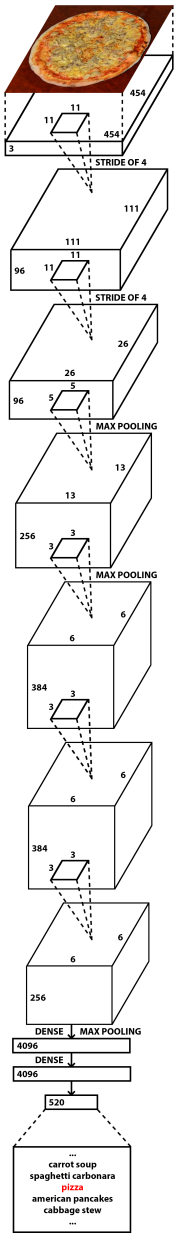

Figure 4. Illustration of the NutriNet architecture used on an image from the recognition dataset with a few example class labels as the output.

We used three different solvers: Stochastic Gradient Descent (SGD) [37], Nesterov's Accelerated Gradient (NAG) [38] and the Adaptive Gradient algorithm (AdaGrad) [39]. All three solvers perform updates on the neural network parameters: SGD performs a parameter update for each training sample,

and both NAG and AdaGrad represent upgrades to this approach. NAG computes the approximation of future parameters, thus gaining the ability to better predict local optima. AdaGrad, on the other hand, adapts the learning rate to the parameters, which means it performs larger updates for infrequent and smaller updates for frequent parameters [40].

The batch size was set according to the layer depth of the deep learning architecture that was being trained, since deeper architectures take up more space on the GPU VRAM; this way, we ensured that we used the largest possible batch size for each architecture. The idea behind this approach is that by filling the GPU VRAM with training images, we minimize the amount of data transfer between the GPU and the CPU RAM or storage drive, which decreases the amount of time the GPU is waiting for new images and thus speeds up the training process. The base learning rate was adjusted with respect to the batch size used, as per Krizhevsky [41]. For AlexNet, a batch size of 256 images and a base learning rate of 0.02 was used; for NutriNet and NutriNet+, 128 images and a rate of 0.01; for GoogLeNet, 64 images and a rate of 0.005; and for ResNet, 16 images and a rate of 0.00125. For the AlexNet, GoogLeNet and ResNet model variants accepting 512×512 pixel images, these values were halved in order to fit them on the GPU VRAM. Additionally, a step-down policy with a step size of 30% and $\gamma = 0.1$ was used for the learning rate of all of the models; these two parameters define the way and speed with which the learning rate decreases over time, with the goal of optimal loss convergence. All of the models were trained in 150 epochs and converged well before the final epoch.

Apart from using dropout, which is implemented in all of the tested deep learning architectures, another technique was used to counter overfitting: the final model was chosen at the training epoch when the loss on the validation subset stops decreasing. This signals the moment when the model stops learning image features that generalize well and instead starts overfitting on the training data. This model was then run on the testing subset once to assess its performance; the resulting accuracy values were used to compare the different deep learning architectures and solvers we tested, which is presented in Section 3.

For model training in the prototype phase, we used three tools: Caffe (NVIDIA's fork of Caffe Version 0.15.9 was used), which is a deep learning framework developed by the Berkeley Vision and Learning Center [42]; the NVIDIA Deep Learning GPU Training System (NVIDIA DIGITS, Version 4.0 was used), which is built upon Caffe and is an interactive deep learning GPU training system that provides a graphical interface and multiple feedback options while training a model [43]; and Torch (Torch Version 7.0 was used), which is a deep learning framework, based on the Lua programming language [44]. Torch was used to train the ResNet models; as such, we used a Torch implementation of ResNet by the Facebook Artificial Intelligence Research team [45]. For all of the other models, we used a combination of Caffe and DIGITS to perform the training. The reason why Torch was used for ResNet model training is that the authors of ResNet used a modified version of Caffe to implement their deep neural network architecture [46], which means that training these models was impossible in the version of Caffe we used. For the online training of NutriNet, described in Section 2.3, only Caffe was used, since the version of the DIGITS GUI we used does not provide scheduling for automatic model training. We trained the models on GPUs because they train deep neural networks up to 13-times faster than CPUs [47]. The GPU that was used in the prototyping phase was an NVIDIA GeForce GTX TITAN X in a local computer and in the online fine-tuning phase an NVIDIA Tesla K80 in a server environment.

2.3. Implementing an Online Training Component

The recognition dataset that we acquired and that is described in Section 2.1 contains 520 classes of food and drink items from various food groups. Despite containing a wide variety of foods and drinks, it still represents only a small subset of all of the available food and drink items, so the aim was to develop a system that would automatically adapt and successfully recognize newly-added food and drink types.

The users, who are Parkinson's disease patients or their carers in our case, classify a new image by taking a photograph with their smartphones. Each time this happens, the photograph is automatically uploaded and saved on our server, and the class label is then provided to the users by the recognition model. Apart from the photograph, its correct class label is also uploaded; the users have the option to correct the label provided by the deep learning model. A Python script is then run on a weekly basis to check whether there are any new images added. If there are, all new images are processed in the same way that the initial recognition dataset was processed, which is described in Section 2.1. If there is a new food or drink class among the newly-uploaded class labels, this new class is added to the dataset. Upon doing that, a Google image search is performed with the new class label as its search query, and new images belonging to this class are added from the web search to complement the user images. This is done so that a newly-added food or drink class contains as many images as possible, which helps to alleviate overfitting. The entire process of adding images from a Google image search, along with the use of the detection deep learning model to remove irrelevant search results is also described in Section 2.1. Finally, the script fine-tunes the deep learning model on this updated image dataset by adjusting the parameters in the neural network.

Caffe is used as the deep learning framework for the online training component, and it uses special "prototxt" files for the definition of both the deep learning architecture and the training parameters (solver type, number of epochs, etc.). One of these files, the one that defines the deep learning architecture for the training process, needs to be updated prior to the fine-tuning process when adding a new class. This consists of changing the number of outputs in the last neural network layer to match the new number of classes and renaming this last layer to force Caffe to relearn it. This is done automatically using the previously-mentioned Python script.

The updated version of the model is then made available on the server to download and perform local image classification. For the purposes of PD Nutrition, the classifications of user images are performed server-side to avoid the need to re-download the model for every new version. Figure 5 illustrates the process of developing and automatically updating the deep learning model.

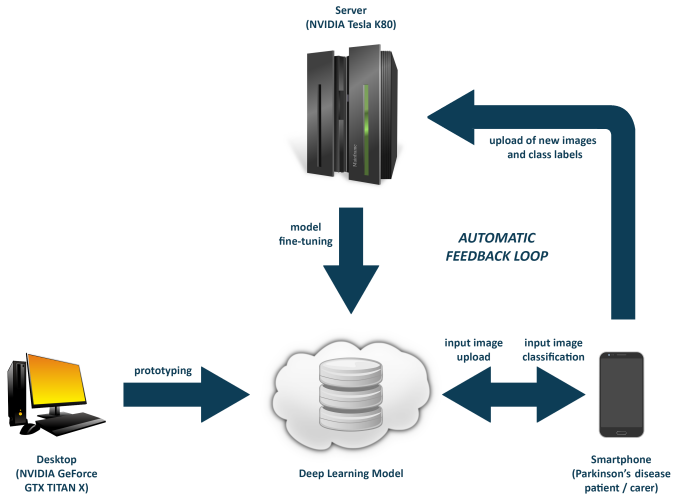

Figure 5. A diagram of the deep learning training process, including the online training component, which keeps the model updated.

3. Results

As was mentioned in Section 2.2, four different deep learning architectures (AlexNet, GoogLeNet, ResNet and NutriNet) and three solver types (SGD, NAG and AdaGrad) were tested for the recognition task; AlexNet, GoogLeNet and ResNet were tested with pre-trained models, as well as with those that accept 512×512 pixel images. NutriNet was additionally tested with an extra convolutional layer,

and this architecture variant is called NutriNet+. Table 1 contains the results for all the tested models; classification accuracy on the testing subset of the recognition dataset (last column in Table 1) was chosen as the main quality measure for the final performance of the models. Figures 6 and 7 contain a visual representation of the accuracy and loss values for all of the models. The pre-trained models with the AdaGrad solver generally performed worse than their SGD and NAG counterparts, whereas the 512 × 512 models mostly performed better with the AdaGrad solver, which seems to indicate that the learning rate selection is more important for 512 × 512 models, as AdaGrad automatically adapts the learning rate to the parameters. The ILSVRC-winning architectures achieved accuracy results according to their layer depths: the deeper the architecture, the better it performed, which is true for pre-trained, as well as 512 × 512 models. When comparing pre-trained and 512 × 512 models using the same architectures, we can see that, on average, the switch to higher-resolution images caused an increase in classification accuracy of 2.53% on the testing subset.

The best-performing model was the 512 × 512 variant of ResNet with the NAG solver, achieving a classification accuracy of 87.96%. NutriNet, on the other hand, achieved its best result with the AdaGrad solver: 86.72%, which is 1.93% higher than its AlexNet counterpart. NutriNet also achieved comparable results to GoogLeNet and was therefore outperformed only by ResNet. When comparing NutriNet to NutriNet+, we can see that the extra convolutional layer did not yield any performance increase, as NutriNet+ models achieved results that are almost identical to the results by NutriNet models. With the exception of ResNet, all models achieved their highest accuracy on the training subset. Finally, 512 × 512 models generally recorded slightly worse results on the validation subset, which is especially true for GoogLeNet, but better results on the testing subset than their pre-trained counterparts, which seems to indicate a drop in the amount of overfitting. For the detection task, NutriNet with the NAG solver was the best-performing model with a classification accuracy of 94.47%.

The training time varied from 11 to 135 h for the recognition models, depending on the deep learning architecture used; ResNet models were by far the most time consuming to train, which is due to the high layer depth of the architecture. The food and drink image detection model was trained for 19 h using the same training parameters as the NutriNet recognition models. All of the reported training times were achieved on the TITAN X GPU. While training is time consuming and computationally expensive, classifying a single image with a deep learning model takes significantly less time, making deployment possible on mobile and web applications.

3.1. Testing NutriNet on Other Datasets

To test how the trained models perform in practice, we built a small testing dataset containing real-world food and drink images. Approximately one-third of the images were taken by us, and two-thirds came from Parkinson's disease patients. The dataset contains 200 images in total, spread across 115 of the 520 classes from our recognition dataset. For testing, we used the best-performing models for each architecture: AlexNet with the AdaGrad solver, GoogLeNet AdaGrad, ResNet NAG and NutriNet AdaGrad, all trained on 512 × 512 pixel images. AlexNet achieved a top-five accuracy of 45%, GoogLeNet 51%, ResNet 58% and NutriNet 55%. The reason we chose to measure the top-five accuracy is the way the model is used in practice: when a user classifies an image, the top five suggestions are provided and the user then chooses the correct one, which is why a top-five accuracy result is more representative of the actual recognition accuracy in practice. Figure 8 contains four distinct examples of images from this real-world dataset and their corresponding output class labels from the NutriNet model: the first image has a correct top-one classification; the second image has an incorrect top-one, but a correct top-five classification; whereas the third image has an incorrect top-five classification. The main reason this image was misclassified is that it contains three different food items, which resulted in inaccurate predictions. The real-world dataset contains more such multi-item images, which decreased the overall classification accuracy of all of the models we tested. While the first three example images contain food items, the fourth image contains a drink item with a correct top-one classification.

Table 1. Results of the deep learning model training on the recognition dataset. SGD, Stochastic Gradient Descent; NAG, Nesterov's Accelerated Gradient; AdaGrad, Adaptive Gradient algorithm; ResNet, Deep Residual Networks.

Model Type	Model	Training Subset		Validation Subset		Testing Subset	
		Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
Pre-Trained Models	AlexNet SGD	0.17	89.35%	0.45	82.87%	0.46	**82.73**%
	AlexNet NAG	0.19	89.32%	0.47	82.76%	0.47	**82.75**%
	AlexNet AdaGrad	0.49	88.33%	0.47	82.31%	0.47	**82.60**%
	GoogLeNet SGD	0.25	90.63%	0.53	83.49%	0.54	**83.91**%
	GoogLeNet NAG	0.31	92.19%	0.54	83.55%	0.53	**83.77**%
	GoogLeNet AdaGrad	0.35	90.62%	0.58	83.53%	0.58	**83.06**%
	ResNet SGD	0.27	84.75%	0.34	85.60%	0.31	**84.82**%
	ResNet NAG	0.34	84.82%	0.40	85.31%	0.35	**85.03**%
	ResNet AdaGrad	0.26	85.23%	0.38	84.14%	0.37	**83.49**%
512×512 Models	AlexNet SGD	0.41	89.76%	0.57	81.98%	0.44	**84.73**%
	AlexNet NAG	0.32	89.89%	0.56	82.03%	0.43	**84.03**%
	AlexNet AdaGrad	0.51	89.33%	0.60	80.20%	0.46	**84.79**%
	GoogLeNet SGD	0.42	90.72%	0.79	80.64%	0.60	**86.39**%
	GoogLeNet NAG	0.35	90.75%	0.78	80.66%	0.58	**86.14**%
	GoogLeNet AdaGrad	0.48	87.50%	0.76	81.22%	0.48	**86.59**%
	ResNet SGD	0.62	81.86%	0.36	85.34%	0.29	**87.76**%
	ResNet NAG	0.45	84.82%	0.29	85.11%	0.26	**87.96**%
	ResNet AdaGrad	0.50	83.76%	0.32	83.91%	0.33	**86.53**%
	NutriNet SGD	0.46	88.59%	0.46	80.81%	0.27	**86.64**%
	NutriNet NAG	0.44	88.53%	0.45	81.06%	0.27	**86.54**%
	NutriNet AdaGrad	0.44	88.76%	0.46	80.77%	0.26	**86.72**%
	NutriNet+ SGD	0.41	88.32%	0.45	81.01%	0.27	**86.51**%
	NutriNet+ NAG	0.45	88.31%	0.45	81.08%	0.27	**86.50**%
	NutriNet+ AdaGrad	0.42	88.35%	0.45	80.88%	0.28	**86.38**%

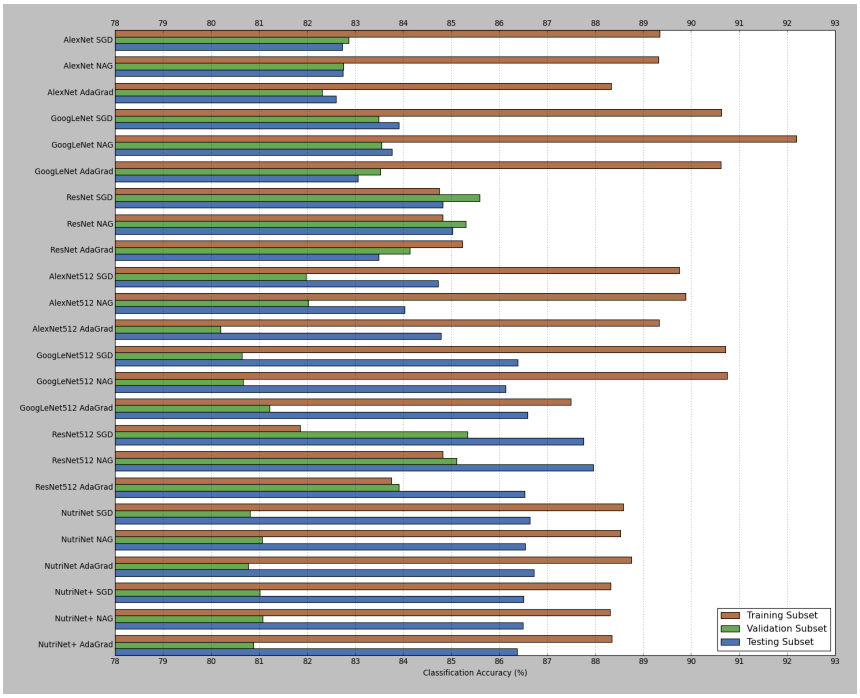

Figure 6. Visual representation of the classification accuracy results from Table 1. The number 512 at the end of some deep learning architecture names indicates a variant of the model that accepts 512×512 pixel images as input.

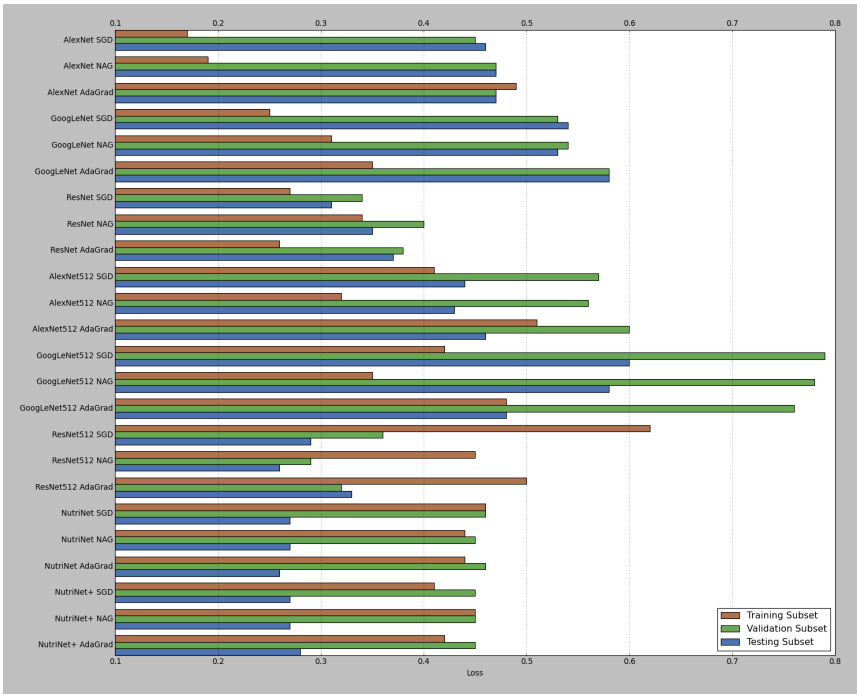

Figure 7. Visual representation of the loss results from Table 1. Similarly to Figure 6, the number 512 indicates a model that accepts 512 × 512 pixel images as input.

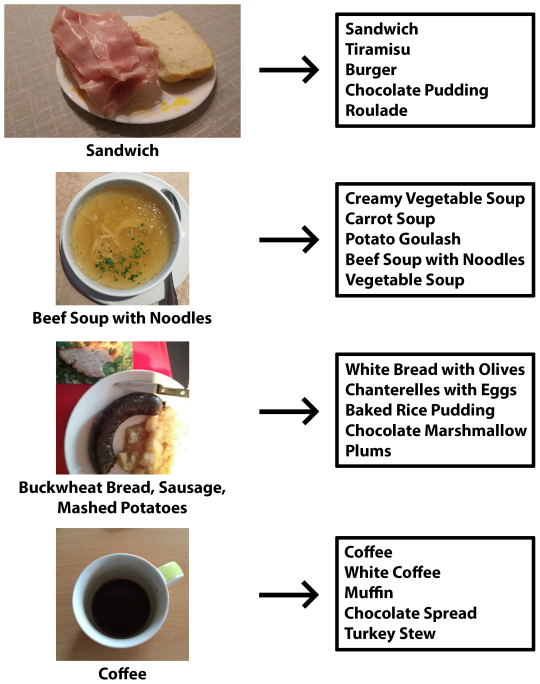

Figure 8. Four example images from the real-world testing dataset and their corresponding class label outputs from the NutriNet model.

To further validate the results that NutriNet achieved on our datasets, we decided to test it on a publicly-available dataset. For this purpose, we chose the most recently-published one: the UNIMIB2016 food image dataset [26]. As was mentioned in Section 1.1, this dataset contains 3616 images of 73 different food items. Furthermore, since these images were collected in an Italian canteen, they contain foods that are closest to the food items present in our datasets, making UNIMIB2016 the most suitable dataset to test NutriNet on. To ensure that our results would be comparable with the baseline results, provided by the authors of the UNIMIB2016 dataset, the dataset was pre-processed as the authors suggested: food classes containing fewer than four instances were removed, leaving 65 classes, and the dataset was split into training and testing subsets. Finally, since NutriNet does not perform image segmentation, the ground-truth bounding-box information that is provided with the dataset was used to crop the food items in the dataset images. Using the NutriNet AdaGrad model, which was pre-trained on our recognition dataset, we performed fine-tuning on the UNIMIB2016 dataset, which took less than an hour. When the authors of the dataset used the ground-truth bounding boxes to segment the food images, they reported a recognition accuracy of 85.80% with their deep convolutional neural network; NutriNet outperformed this result, as it achieved an accuracy of 86.39% on the UNIMIB2016 dataset.

4. Discussion

The main result of our research is two-fold: the newly-defined NutriNet deep convolutional neural network architecture and the food and drink image recognition dataset, which contains a much larger number of different food types than previous efforts in the field [4,6–8,14–18,20,21,23,24,26,27] and, unlike these works, also contains a wide variety of drinks. Furthermore, since all of the images from our recognition dataset are freely available online, the dataset can be replicated by other researchers and even tailored to food and drink items local to their area. To facilitate this process, the tools we used to download images from the recognition dataset were made available online on the Jožef Stefan Institute website [33] and in the Supplementary Materials.

An additional difference between our solution and the majority of previous research is that our food and drink image recognition system is being used in practice for the dietary assessment of Parkinson's disease patients. The accuracy results of NutriNet, presented in Section 3, are also very promising and encouraging. We achieved a classification accuracy of 86.72% for the recognition task, which is higher than the accuracy values reported by most of the other deep convolutional neural network approaches in the field [14,18,20,21,23,24,26]. The detection model achieved an accuracy of 94.47%, which is comparable to the detection results reported by other researchers [18,23]. However, since testing was performed on different datasets in these studies, the results are not directly comparable with ours. On the other hand, testing on the publicly-available UNIMIB2016 dataset showed that NutriNet outperforms the baseline method provided with the dataset [26]. Additionally, it is important to note that the classification accuracy generally decreases with the increase in the number of classes in the dataset, which makes our results even more encouraging, given that the number of classes in our recognition dataset far exceeds the solutions mentioned above.

We attribute the better results of NutriNet compared to the AlexNet deep learning architecture to the fact that it is able to gain additional knowledge from the input images as it learns a more complex representation of the input images. NutriNet achieved results on our recognition dataset that are comparable to the results by GoogLeNet, and of the tested architectures, only ResNet outperformed it. When comparing the classification accuracy of the architectures on the real-world dataset, we can observe that their order is the same as on the recognition dataset: from the lowest-performing AlexNet, to GoogLeNet and NutriNet and, finally, ResNet. However, it is important to note that NutriNet models are considerably faster to train than 512×512 ResNet models, with a training speed of about five epochs per hour as opposed to ResNet's one epoch per hour (AlexNet's and GoogLeNet's training speeds with 512×512 pixel images are also slower, with 3.5 and one epoch per hour, respectively), which is mainly due to the fact that reduced image batch sizes have to be used with the deeper architectures. AlexNet, on the other hand, is slower than NutriNet because it accepts a larger number

of parameters. This makes the NutriNet architecture viable for settings where training time is an important part of the problem, such as in our case, where models are continually fine-tuned on a weekly basis.

Due to the complexity of food and drink images, many of the previously-proposed methods for food recognition achieved a low classification accuracy, and drink image recognition methods were previously nonexistent. This is where deep learning comes in. Food and drink items have features that are difficult to define, making automated feature definition a more appropriate approach. The results of our research work further confirm this. However, despite using the dropout technique and selecting the model at the point in the training process when the validation loss stops decreasing, overfitting remains a problem with deep learning. In our case, the issue is that there are many different classes of food and drink items, and because the classes are unbalanced, the rarer classes generate fewer images, which introduces a greater risk of overfitting on the few images of that class that are in the dataset.

Overfitting could also be one of the reasons why the classification accuracy is lower on real-world images than on images from the testing subset, with other possible reasons being added noise and occlusion in real-world images and the fact that our recognition dataset could still contain some irrelevant images: the dataset was cleaned with a food and drink image detection model that has an accuracy of 94.47%, which means that the vast majority of images are correctly classified in the recognition dataset, but not necessarily all of them. As a consequence, this could lower the classification accuracy for real-world images. Finally, since we do not perform image segmentation, irrelevant items present in the images make the recognition task more challenging.

A shortcoming of our food and drink recognition system is that the deep learning model is limited to one output per image, which means that not every item gets successfully recognized in images with multiple food or drink items; an example of such an image is the third real-world image in Figure 8. This is true for all of the tested models and is another reason they performed worse on the real-world dataset than on the recognition dataset. In the current state of the recognition system, we are classifying 520 different food and drink items. While that number is considerably higher than in publicly-available datasets in the field [4,15–17,26], it is still limited when compared to the number of all of the possible foods and drinks. We address this issue by automatically adding new classes to the dataset from the class labels users provide when trying to classify a new image.

5. Conclusions

In this paper, we present the food and drink image detection and recognition system that we built, in the scope of which we developed a deep convolutional neural network architecture called NutriNet in order to provide a higher classification accuracy for the recognition of food and drink images from the 520-class dataset that we acquired using Google image searches, while keeping the model training time low to enable faster fine-tuning. Our recognition system is used inside the PD Nutrition dietary-assessment application for Parkinson's disease patients, and it also incorporates online training that automatically updates the model with new images and new food and drink classes.

The next step in our research will be to further modify the NutriNet architecture, which performed well, but there is still room for improvement, especially on real-world images with added noise and obstructions. Since there are many possibilities to alter the architecture, we will be looking to implement optimization methods to automate this step, as well. As additional food and drink images are added automatically to the dataset by Parkinson's disease patients, we also hope to further address the problem of overfitting. Additionally, to classify images with multiple food or drink items, a food and drink detection model could be trained. Each of the outputs of this model would represent a separate food or drink item that could then be used as the input to the existing recognition model. Another approach to this problem would be to join the detection and recognition steps and perform both in a single deep convolutional network; further testing would then be required to determine which of these approaches would yield better results for the final goal of food and drink image recognition.

Supplementary Materials: The tools needed to download images from the food and drink image recognition dataset detailed in this paper are available online at http://www.mdpi.com/2072-6643/9/7/657/s1 and they include the following files: "download_images.py" is a Python script used to download images from Google image search queries; "readme.txt" is a guide describing the necessary steps to obtain the food and drink image dataset using the aforementioned script; "slo_foods_drinks.txt" is a complete list of all of the food and drink class labels we used to build the recognition dataset (the list is in Slovene).

Acknowledgments: This work is supported by the project mHealth Platform for Parkinson's Disease Management (PD_manager), which received funding from the European Union's Horizon 2020 Research and Innovation program under Grant Number 643706. This work is also supported by the project Research Infrastructure on Consumer Health and Food Intake using E-Science with Linked Data Sharing (RICHFIELDS), which received funding from the European Union's Horizon 2020 Research and Innovation program under Grant Number 654280. The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding Number P2-0098). The authors would also like to thank Drago Torkar for providing feedback regarding the computer vision aspects of this research work, and Javier de la Cueva for providing advice regarding web image copyrights.

Author Contributions: Simon Mezgec and Barbara Koroušić Seljak conceived of and designed the research work, provided real-world testing data and analyzed the data. Simon Mezgec obtained the image datasets, trained and tested the neural networks, developed the online training component and wrote the manuscript. Barbara Koroušić Seljak edited the manuscript. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444, doi:10.1038/nature14539.
2. Hubel, D.H.; Wiesel, T.N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *J. Physiol.* **1962**, *160*, 106–154, doi:10.1113/jphysiol.1962.sp006837.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the NIPS'12, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
4. Chen, M.; Dhingra, K.; Wu, W.; Yang, L.; Sukthankar, R.; Yang, J. PFID: Pittsburgh Fast-Food Image Dataset. In Proceedings of the ICIP 2009, Cairo, Egypt, 7–10 November 2009; pp. 289–292.
5. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the ICCV'99, Corfu, Greece, 20–21 September 1999; pp. 1150–1157.
6. Joutou, T.; Yanai, K. A Food Image Recognition System with Multiple Kernel Learning. In Proceedings of the ICIP 2009, Cairo, Egypt, 7–10 November 2009; pp. 285–288.
7. Yang, S.; Chen, M.; Pomerlau, D.; Sukthankar, R. Food Recognition using Statistics of Pairwise Local Features. In Proceedings of the CVPR 2010, San Francisco, CA, USA, 13–18 June 2010; pp. 2249–2256.
8. Anthimopoulos, M.M.; Gianola, L.; Scarnato, L.; Diem, P.; Mougiakakou, S.G. A Food Recognition System for Diabetic Patients Based on an Optimized Bag-of-Features Model. *JBHI* **2014**, *18*, 1261–1271, doi:10.1109/jbhi.2014.2308928.
9. GoCARB—Type 1 Diabetes Self-Management and Carbohydrate Counting: A Computer Vision Based Approach. Available online: http://www.gocarb.eu (accessed on 30 December 2016).
10. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the CVPR 2015, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
11. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv* **2016**, arXiv:1606.02147.
12. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the ECCV'16, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
14. Kawano, Y.; Yanai, K. Food Image Recognition with Deep Convolutional Features. In Proceedings of the UbiComp 2014, Seattle, WA, USA, 13–17 September 2014; pp. 589–593.
15. Matsuda, Y.; Hoashi, H.; Yanai, K. Recognition of Multiple-Food Images by Detecting Candidate Regions. In Proceedings of the ICME 2012, Melbourne, Australia, 9–13 July 2012; pp. 25–30.
16. Kawano, Y.; Yanai, K. Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation. In Proceedings of the ECCV'14, Zürich, Switzerland, 6–12 September 2014; pp. 3–17.

17. Bossard, L.; Guillaumin, M.; Van Gool, L. Food-101—Mining Discriminative Components with Random Forests. In Proceedings of the ECCV'14, Zürich, Switzerland, 6–12 September 2014; pp. 446–461.
18. Kagaya, H.; Aizawa, K.; Ogawa, M. Food Detection and Recognition using Convolutional Neural Network. In Proceedings of the MM'14, Orlando, FL, USA, 3–7 November 2014; pp. 1055–1088.
19. ImageNet. Available online: http://image-net.org (accessed on 30 December 2016).
20. Yanai, K.; Kawano, Y. Food Image Recognition using Deep Convolutional Network with Pre-Training and Fine-Tuning. In Proceedings of the ICMEW 2015, Turin, Italy, 29 June–3 July 2015; pp. 1–6.
21. Christodoulidis, S.; Anthimopoulos, M.M.; Mougiakakou, S.G. Food Recognition for Dietary Assessment using Deep Convolutional Neural Networks. In Proceedings of the ICIAP 2015, Genoa, Italy, 7–11 September 2015; pp. 458–465.
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the CVPR 2015, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
23. Singla, A.; Yuan, L.; Ebrahimi, T. Food/Non-Food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. In Proceedings of the MADiMa'16, Amsterdam, The Netherlands, 15–19 October 2016; pp. 3–11.
24. Liu, C.; Cao, Y.; Luo, Y.; Chen, G.; Vokkarane, V.; Ma, Y. DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. In Proceedings of the ICOST 2016, Wuhan, China, 25–27 May 2016; pp. 37–48.
25. Tanno, R.; Okamoto, K.; Yanai, K. DeepFoodCam: A DCNN-Based Real-Time Mobile Food Recognition System. In Proceedings of the MADiMa'16, Amsterdam, The Netherlands, 15–19 October 2016; p. 89.
26. Ciocca, G.; Napoletano, P.; Schettini, R. Food Recognition: A New Dataset, Experiments, and Results. *JBHI* **2017**, *21*, 588–598, doi:10.1109/jbhi.2016.2636441.
27. Hassannejad, H.; Matrella, G.; Ciampolini, P.; De Munari, I.; Mordonini, M.; Cagnoni, S. Food Image Recognition using Very Deep Convolutional Networks. In Proceedings of the MADiMa'16, Amsterdam, The Netherlands, 15–19 October 2016; pp. 41–49.
28. OPEN—Platform for Clinical Nutrition. Available online: http://opkp.si/en_GB (accessed on 30 December 2016).
29. PD_manager—mHealth Platform for Parkinson's Disease Management. Available online: http://parkinson-manager.eu (accessed on 30 December 2016).
30. Google Custom Search. Available online: http://developers.google.com/custom-search (accessed on 30 December 2016).
31. Python Software Foundation. Available online: http://www.python.org/psf (accessed on 24 June 2017).
32. Symas Lightning Memory-Mapped Database. Available online: http://symas.com/lightning-memory-mapped-database (accessed on 28 May 2017).
33. NutriNet Food and Drink Image Recognition Dataset Tools. Available online: http://cs.ijs.si/korousic/imagerecognition/nutrinetdatasettools.zip (accessed on 5 June 2017).
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR* **2014**, *15*, 1929–1958.
36. Howard, A.G. Some Improvements on Deep Convolutional Neural Network Based Image Classification. *arXiv* **2013**, arXiv:1312.5402.
37. Bottou, L. Large-Scale Machine Learning with Stochastic Gradient Descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; pp. 177–186.
38. Nesterov, Y. A Method of Solving a Convex Programming Problem with Convergence Rate $O(\frac{1}{k^2})$. *Sov. Math. Dokl.* **1983**, *27*, 372–376.
39. Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR* **2011**, *12*, 2121–2159.
40. Ruder, S. An Overview of Gradient Descent Optimization Algorithms. *arXiv* **2016**, arXiv:1609.04747.
41. Krizhevsky, A. One Weird Trick for Parallelizing Convolutional Neural Networks. *arXiv* **2014**, arXiv:1404.5997.
42. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the MM'14, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.

43. NVIDIA DIGITS—Interactive Deep Learning GPU Training System. Available online: http://developer.nvidia.com/digits (accessed on 30 December 2016).
44. Torch—A Scientific Computing Framework for LuaJIT. Available online: http://torch.ch (accessed on 8 April 2017).
45. Torch Implementation of ResNet and Training Scripts GitHub. Available online: http://github.com/facebook/fb.resnet.torch (accessed on 7 April 2017).
46. Deep Residual Learning for Image Recognition GitHub. Available online: http://github.com/KaimingHe/deep-residual-networks (accessed on 16 April 2017).
47. GPU Technology Conference 2015—Deep Learning on GPUs. Available online: http://on-demand.gputechconf.com/gtc/2015/webinar/deep-learning-course/intro-to-deep-learning.pdf (accessed on 30 December 2016).

Chapter 3

Mixed Deep Learning and Natural Language Processing Method for Fake-Food Image Recognition and Standardization to Help Automated Dietary Assessment

The work, presented in Chapter 2, is limited to one food or drink item per image. This is appropriate for simpler implementations of food image recognition solutions, but it requires additional effort by the individual as it is necessary to take images that contain only one food item. Since real-world images of foods and drinks rarely contain just one item, and because the goal of automating dietary assessment is to reduce effort, this approach is not optimal for implementing a practical application for dietary assessment.

This is why the next step in researching food image recognition was to develop a solution that would not be limited to any number of food items per image. The biggest issue with implementing such an approach is that there were no publicly available large-scale image datasets at the time that would have multiple food or drink items annotated per image, and building a dataset like that with web searches is also impossible.

Due to this lack of suitable image datasets of real food, a fake-food image recognition system was developed. Food replicas are used to study meal composition, food choice, and other behavioral aspects of diets [27]. Crucially, researchers take images of participants' food choices. Because these replicas are visually very similar to real food, and because fake-food images often contain a large number of food and drink items, they are well suited for the recognition of multiple food and drink items per image. A fake-food image dataset was sourced from the fake food buffet (FFB) research method [27] and annotated manually. Similarly to prior work, data augmentation steps were performed on the dataset.

The FFB dataset was used to develop a solution that is able to segment fake-food images on a pixel level. This solution was implemented by using an existing DCNN approach—fully convolutional networks (FCNs) [28]. Specifically, the FCN-8s version [28] was used to train a model on the FFB dataset, as this version is capable of segmenting the image at the finest grain. The results of the training and testing process are included in the publication below.

Permission to include the publication *"Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment"* [17] in this doctoral dissertation was confirmed by the journal *Public Health Nutrition* in an email exchange from 7 June 2021.

Public Health Nutrition: 22(7), 1193–1202 doi:10.1017/S1368980018000708

Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment

Simon Mezgec[1,2], Tome Eftimov[1,2], Tamara Bucher[3,4] and Barbara Koroušić Seljak[2,*]
[1]Jožef Stefan International Postgraduate School, Ljubljana, Slovenia: [2]Computer Systems Department, Jožef Stefan Institute, Jamova cesta 39, Ljubljana 1000, Slovenia: [3]Institute of Food, Nutrition and Health (IFNH), ETH Zürich, Zürich, Switzerland: [4]School of Health Sciences, Faculty of Health and Medicine, Priority Research Centre in Physical Activity and Nutrition, The University of Newcastle, Callaghan, Australia

Submitted 31 July 2017: Final revision received 23 February 2018: Accepted 27 February 2018: First published online 6 April 2018

Abstract

Objective: The present study tested the combination of an established and a validated food-choice research method (the 'fake food buffet') with a new food-matching technology to automate the data collection and analysis.

Design: The methodology combines fake-food image recognition using deep learning and food matching and standardization based on natural language processing. The former is specific because it uses a single deep learning network to perform both the segmentation and the classification at the pixel level of the image. To assess its performance, measures based on the standard pixel accuracy and Intersection over Union were applied. Food matching firstly describes each of the recognized food items in the image and then matches the food items with their compositional data, considering both their food names and their descriptors.

Results: The final accuracy of the deep learning model trained on fake-food images acquired by 124 study participants and providing fifty-five food classes was 92·18 %, while the food matching was performed with a classification accuracy of 93 %.

Conclusions: The present findings are a step towards automating dietary assessment and food-choice research. The methodology outperforms other approaches in pixel accuracy, and since it is the first automatic solution for recognizing the images of fake foods, the results could be used as a baseline for possible future studies. As the approach enables a semi-automatic description of recognized food items (e.g. with respect to FoodEx2), these can be linked to any food composition database that applies the same classification and description system.

Keywords
Fake food buffet
Food replica
Food image recognition
Food matching
Food standardization

Measuring dietary behaviour using traditional, non-automated, self-reporting technologies is associated with considerable costs, which means researchers have been particularly interested in developing new, automated approaches. There is a clear need in dietary assessment and health-care systems for easy-to-use devices and software solutions that can identify foods, quantify intake, record health behaviour and compliance, and measure eating contexts. The aim of the present study was to test the combination of an established and validated food-choice research method, the 'fake food buffet' (FFB), with a new food-matching technology to automate the data collection and analysis.

The FFB was developed as an experimental method to study complex food choice, meal composition and portion-size choice under controlled laboratory conditions. The FFB is a selection of very authentic replica-food items, from which consumers are invited to choose. The FFB method was validated by a comparison of meals served from real and fake foods[1]. The food portions served from the fake foods correlated closely with the portions served from the real foods[1]. Furthermore, significant correlations between the participants' energy needs and the amounts served were found in several studies[1–4]. It has also been shown that people who selected foods for an entire day from an FFB were able to closely match their dietary requirements[5].

In a typical FFB study, the experimenters choose fake foods and set up a buffet. The participants receive instructions, which can contain the experimental intervention, and are then invited to select foods, choose portions of foods to assemble meals[2,3] or even set a diet for a day[5]. The experimenter then analyses the choice. Similar protocols and the same fake foods were

*Corresponding author: Email barbara.korousic@ijs.si

used for experiments in different countries (i.e. Germany, Switzerland, the UK and Australia). Currently, the FFB study procedure still has several 'analogue' components. After the participants select the meals, a photograph is taken, the foods are separated manually, each food is weighed, and the researcher calculates the nutritional values for the selected fake foods. This process would benefit from automation. All the consumer choices are recorded and additional fake-food images are available for the aims of the research.

The first step of the automation process is to recognize the fake-food and fake-drink items present in these images. Due to the nature not only of the fake-food and fake-drink items, but also of food and drink items in general, this is a particularly challenging computer vision problem. Differentiating between different food or drink items (henceforth 'food items') can sometimes be challenging even for the human eye. The issue is that different food items can appear to be very similar and the same food item can appear to be substantially different on different images because of a variety of factors, such as image quality, illumination, the amount of noise present in the image, the way in which the food item was prepared and served, etc.

The next step is to match the fake-food items recognized in the image to food composition data, which are detailed sets of information on the nutritionally important components of foods, providing values for the energy and nutrients, including protein, carbohydrates, fat, vitamins and minerals, and for other important food components, such as fibre, etc. The data are presented in food composition databases (FCDB). The process of semi-automatic food matching is a crucial part of an automated dietary assessment.

In the current paper, we present results of a study performed with the objective to develop an automated dietary assessment that consists of two main activities: (i) automatically recognizing fake-food and fake-drink items from photos; and (ii) automatically assigning (matching) recognized items to their compositional data. Using this approach, the dietary assessment can be performed much more quickly and, in many cases, also more accurately than if performed manually.

The paper proceeds as follows. In the next section we present relevant work on the FFB, food image recognition and food matching. Thereafter we introduce the methodology applied in the present study to an automated dietary assessment. Next we show how this methodology was applied to fake foods and present the results of the evaluation. Finally, we discuss the results and present some ideas for future work.

Relevant work

The fake food buffet

Replica-food models such as the Nasco food models[6] have traditionally been used in dietary assessment as portion-size estimation aids and for educational purposes. However, only recently have food-replica models been validated and used for experimental studies in food-choice and consumer behaviour research[1]. The FFB method has, for example, been used to investigate environmental influences such as plate size[3], vegetable variety[7,8] in food choice, or the effect of the nutritional information and labels on food choice for a single meal[2,9] or for an entire day[5]. Fake foods were also used to investigate health perceptions[4,10] and social influences and attitudes to food choices[11,12].

Meanwhile, the FFB is an established research tool within several research facilities worldwide; research institutions in Germany, Switzerland, the UK and Australia are using a similar set of replica foods to address a variety of research questions. However, to date the procedure of carrying out an FFB experiment still involves several manual steps, including identifying and quantifying the foods selected by the study participants, and different research laboratories use different FCDB to calculate the theoretical nutrient contents of the fake foods. The differences in the nutrient profile of the same food between different nutrient databases in different countries might reflect actual differences in the composition of these foods in the different countries. Linking the fake foods to standardized nutrient contents (e.g. an EU database) might remove certain country-specific information (e.g. related to food processing). However, the standardization of the nutrient content calculation would still greatly facilitate international collaboration and the comparison of food portions.

Food image recognition

Until recently, the approach favoured by most researchers in the field of food image recognition was based on manually defined feature descriptors[13–15]. However, because of the complexity of the features in food images, this approach did not perform well.

Recently, deep learning, a fully automatic machine learning approach, achieved state-of-the-art results in a wide variety of computer vision problems and proved to be most effective for the task of image recognition. It has also been validated in the field of food image recognition multiple times[16–23]. However, to the best of our knowledge, there are no previous solutions that would automatically recognize drinks from images, and the number of food classes in the data sets that have been used so far is very limited – often up to 100 different food types or less. This is why we have introduced an approach that addresses both of these issues[24]. It is a unique approach due to how the food and drink image data set is built as well as the custom deep learning network used. Using this approach, we have achieved an accuracy of 86·72 % on a new data set containing 520 different food and drink items. However, our approach, as well as most solutions listed above, have a shortcoming: they are incapable of

recognizing more than one food item per image. We address this issue in the current paper as we are performing pixel-level classification, which is not limited to any specific number of recognized food items.

The research works described above classify food items into food classes, which can then be linked to FCDB to add compositional information. However, there is another approach to this problem: perform food ingredient recognition and try to directly recognize the food ingredients from the image. This has been presented in a few recent solutions by Chen *et al.*[25,26] and Salvador *et al.*[27], which detail the process of recognizing ingredients from food images and then linking them with recipes containing those ingredients.

Food matching

Matching food items with compositional data can be performed in two ways, by considering either the food descriptors or the food names. Databases on food composition, consumption, allergens, etc. describe food items with descriptors (terms and facets) defined by a classification and indexing system. Several such systems exist (e.g. FoodEx2[28], LanguaL[29]); however, many databases are lacking food descriptors because defining them is a time-consuming task. Therefore, matching food items from different data sources by considering food names is a relevant challenge. The problem of matching food with compositional data through food names is that the same food can have different food names within different data sources (i.e. different FCDB)[30]. This is because people who express themselves in different ways or have unique writing styles defined the food names. For example, the food item name that results from the food image recognition method depends on the person who developed the method, while the food item name presented in the FCDB depends on the person or company who performed the nutrient analysis and then provided and stored the result. To address this problem, in 2016 we developed a promising method for matching food items to their compositional data using food names and text-similarity measures applied at a word level, which was aimed at matching food items to their compositional data[31]. Meanwhile, we have extended this method to classify and describe food items considering both food names and food descriptors that are semi-automatically assigned to the food items[32].

Methods

The fake food buffet

In the current study we used the image data from an FFB experiment in which 124 participants were invited to serve themselves lunch from a buffet with replica foods. Details about the procedures of the experimental study are described elsewhere[2]. In total, 121 photographs were used (two images were missing, one image was incomplete) and out of the fifty-seven food classes, fifty-five were matched ('margarine' was not present in any images and 'fish sticks' were present in only one image, which is not enough to train a deep learning model).

Fake-food image recognition

Food image recognition requires several steps to be performed: image pre-processing, deep learning model training, testing and validation. We are also performing data augmentation in the pre-processing step, by which we are referring to the process of expanding the original image data set by generating additional variants of original images, which is beneficial for deep learning methods as they require as large a data set as possible for increased real-world accuracy[33].

Image pre-processing

To train a deep learning model on the fake-food images we first needed to manually pre-process the images. The main aim of the pre-processing step is to generate 'ground-truth' labels for the food items present in each image, which are later needed for the supervised learning of the deep learning model. Ground truth refers to information that we know is correct; in the case of food images, this means that the labels for each of the food items are reliable. Usually, the simplest approach to generating such labels is labelling each image with one food class (food name) and training a deep learning model in such a way that it returns one text label per image. However, since all the images from the FFB not only contain multiple food items, but have over eleven foods on average, such an approach would be very inaccurate and is therefore not appropriate for this application.

That is why for generating ground-truth data we needed to label not just each image, but each food item present in each image.

As foods often overlap on plates and drinks can obstruct the view of other items, we labelled each food item on a pixel level, which means that the result of this step was a new label image with the same width and height as the input image, only with a single channel as opposed to three channels used in RGB images. This label image contains a class prediction for each individual pixel, so a 'tomato' item has all its pixels labelled as 'tomato' and its surrounding pixels are labelled as another class.

Since generating such ground-truth labels without significant errors is non-trivial and is one of the main obstacles when trying to design a pixel-level classification solution, we manually segmented each food and drink item in each of the 121 fake-food images. This has resulted in 121 label images with a total of 1393 different food and drink items, each belonging to one of the fifty-five food and drink classes.

After the labelling part, the fake-food data set was randomly split into training (70 % of images), validation (10 %) and testing (20 %) subsets to use for the deep

learning model training such that any image was used in only one of the subsets. The food objects are the same across all three subsets, although the selection of food objects differs from image to image. Finally, four different data-augmentation steps were performed on the images in the training subset, as well as their corresponding label images. These steps included: rotating each image by 90°, 180° and 270°; flipping the image horizontally; adding random colour noise; and zooming in on the image so that 25 % of the image's borders were removed[24]. It is important to note that while the other data-augmentation steps were performed in the same way on both the fake-food images and the label images, random noise was introduced only to the food images, as the ground-truth labels should not change, even in the presence of noise. The result of the data-augmentation process is therefore seven variations per fake-food image in the training subset. In total, the final fake-food data set with the augmented training subset contains 631 images with 7222 food or drink items (some items were cut off in the zoomed-in image variants). All the fake-food and label images have a resolution of 500 pixels × 375 pixels; the reason for the lower resolution is the considerable memory requirements of the deep learning approach used, which is described in the following section.

Deep learning model training

We trained the fake food and drink recognition model using deep convolutional neural networks, which are a type of neural network that works in a similar way to human vision: individual neurons react to overlapping regions in the visual field. Specifically, we used fully convolutional networks (FCN) that were introduced in a study by Long *et al.*[34] and represent the state-of-the-art for semantic segmentation. This process segments the input image into separate parts and then classifies each part into an output class; the network does that by performing pixel-level classification. The FCN therefore outputs a pixel map instead of a class text label, and this pixel map contains predictions from the model for each individual pixel of the input image, as opposed to having only one prediction for the entire image. This is important because, as mentioned in the previous section, it is the most accurate way to describe all the food items present in one image. Long *et al.*[34] introduced three FCN variants: FCN-32s, FCN-16s and FCN-8s. The FCN-32s outputs a pixel map based on the predictions from the final layer of the fully convolutional network, which is the standard approach for semantic segmentation networks. The FCN-16s, on the other hand, combines the predictions from the final layer with those from an earlier layer, which contains a more detailed representation of the input image, thus allowing the network to make predictions at a finer grain. Finally, the FCN-8s considers an additional layer when making predictions compared with the FCN-16s, and it is therefore able to segment the input images at the finest grain. This is why, of all the FCN variants available, the FCN-8s is the best performing, making it suitable for food and drink image recognition.

Since it is possible to use deep learning models that are pre-trained on other data sets as a starting point for the model training process, we wanted to use an FCN-8s model that was pre-trained on the PASCAL Visual Object Classes (PASCAL VOC) data set[35] to decrease the training time and increase the number of images for training, thus improving the robustness of the final model. However, since this data set contains images from only twenty-one different classes, we needed to modify the FCN-8s network architecture to use it for the recognition of our fifty-six classes (fifty-five fake-food classes and the background class). This was done by adding an extra layer at the end of the deep learning network, which increases the number of output classes from twenty-one to fifty-six. Doing this was necessary to take advantage of the pre-trained network, as otherwise the output layer would have to be retrained from the start.

For the deep learning model training we used the popular deep learning framework Caffe, which was developed by the Berkeley Vision and Learning Center[36], and the NVIDIA Deep Learning GPU Training System (NVIDIA DIGITS), which is a graphical user interface built upon Caffe and provides feedback options during the model training process[37].

To train the models, we used Adam[38] as the solver. Solvers are methods that perform updates to deep neural network parameters in each training epoch with the goal to minimize the loss function, which is the primary quality measure while training the models. The solver is therefore an important part of the deep learning model training process that tunes the model in such a way that it reacts to features in the input images and learns to classify them successfully. Adam is a solver that automatically adapts the learning rate to the parameters. The learning rate defines the rate with which the parameters are changed during the training process; the higher the learning rate, the faster the model converges to the optimal loss value, which speeds up the training. However, the learning rate should not be set too high because the model might then converge to a worse loss value, or not converge at all. It is therefore important to choose an appropriate rate, and we achieved the best results by setting the initial learning rate to 0·0001 and letting Adam automatically adapt this rate during the training.

Since the FCN perform the classification of each individual pixel, their memory requirements are much greater than those of traditional convolutional neural networks where large batches of images can be processed at the same time. Because of this we had to set the software to process only one image at a time, as one image alone completely filled the video random access memory of the graphics processing unit. Additionally, we trained the model for 100 epochs and then selected the final model at the epoch where the loss on the validation subset stopped

decreasing, as that signals the moment when the model starts overfitting on the training data. For the model training, we used a single NVIDIA GeForce GTX TITAN X graphics processing unit.

Measures

To measure the performance of the trained deep learning model we used the same evaluation measures as Long *et al.*[34], since their study showed that these measures are appropriate to test the FCN models. The measures are based on the standard pixel accuracy and Intersection over Union (IU) measures, including the following.

$$\text{Pixel accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i},$$

$$\text{Mean accuracy} = \frac{1}{n_{\text{cl}}} \sum_i \frac{n_{ii}}{t_i},$$

$$\text{Mean IU} = \frac{1}{n_{\text{cl}}} \sum_i \frac{n_{ii}}{\left(t_i + \sum_j n_{ji} - n_{ii}\right)}$$

and

$$\text{Frequency-weighted IU} = \left(\sum_k t_k\right)^{-1} \sum_i \frac{t_i n_{ii}}{\left(t_i + \sum_j n_{ji} - n_{ii}\right)},$$

where n_{cl} is the number of different classes in the ground-truth labels, n_{ij} is the number of pixels of class i predicted to belong to class j and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i in the ground-truth labels. We used a Python implementation of these measures[39].

Food matching

To match the food items recognized in the image to an FCDB, we decided to use an approach that involved matching foods by their descriptors and names to achieve the best possible result. However, because most FCDB are lacking food descriptors, we first applied the StandFood method[32] to assign FoodEx2 descriptors to the food items in a semi-automated way.

The StandFood method consists of three parts. The first identifies what type of food (raw, derivative, simple or aggregated composite food) is being analysed. This is the classification part that involves a machine learning approach[40]. The second part describes the food using natural language processing[41] combined with probability theory, which results in the list term or FoodEx2 code for the food. For each food item that needs to be described according to FoodEx2, its English name is used. The name is pre-processed by converting it to lowercase letters. Part-of-speech (POS) tagging is used to extract its nouns, adjectives and verbs. The extracted sets are further transformed using lemmatization. Using the extracted nouns, the FoodEx2 data are searched for the names that consist of at least one of the extracted nouns. The resulting list (a subset) is then pre-processed by converting each food item's name to lowercase letters, applying POS tagging to extract the nouns, adjectives and verbs, and using lemmatization for the extracted sets. Then, the food item that needs to be described according to FoodEx2 is matched with each food item in the resulting list and a weight is assigned to each matching pair. Finally, the pair with the highest weight is the most relevant one, so it is returned together with its food category from FoodEx2. The third part combines the result from the first and the second part by defining post-processing rules to improve the result for the classification part.

The first evaluation of the system was made using 532 foods from the Slovenian FCDB and had an accuracy of 89 % for the classification part and 79 % for the description part. However, 21 % of instances were not correctly described, even though some of these instances were correctly classified. This happens due to the fact that the food items do not exist in FoodEx2, the food items are specific to some cultures, or the POS tagging model that is used for the extraction of the morphological information does not provide nouns, so the search cannot continue.

For the purposes of the current study we extended the StandFood method in the second part. The extension works with cases of food names where nouns cannot be extracted, so instead of using the POS tagging-probability-weighted method[42] to find the most relevant match, it switches to the Levenshtein distance[43], which can be used as a similarity measure between two textual descriptions.

The methodology

Figure 1 shows a flowchart of the methodology applied in the present study. First, the food image recognition process uses a fake-food image to find the classes (names) of all the food items in the image. These food names are then processed by the StandFood method to define the FoodEx2 descriptors of the recognized food items. Once both the food names and the descriptors are identified, the recognized fake foods can be matched with compositional data from the FCDB. The final result is therefore a fake-food image standardized with unique descriptors, which enables the conversion of food intake into nutrient intake and helps the automated dietary assessment.

Experimental results

Results from food image recognition

The training of the FCN-8s deep learning model took approximately 37 h of computation on the previously mentioned graphics processing unit. Classifying a single image, however, takes significantly less time and computing power, which makes the use of deep learning models possible even in mobile applications. After the

training was completed using the training and validation subsets, the model was run once on the testing subset. This generated label images for the fake-food images, which were then compared with the ground-truth label images using the measures mentioned above. Table 1 contains these results, whereas Fig. 2 contains three example images (one from each subset) with the corresponding ground-truth and model prediction labels.

As expected, the performance of the FCN-8s model was better on the training subset than on the other two subsets. However, the difference is not substantial, which means the model learned features that generalize well. It is important to note that this performance was measured on all classes; this includes the background, which represents the majority of the pixels. Since the testing subset contains images new to the deep learning model, we consider the results on this subset to be the most representative of real-world performance. Out of these results, we chose pixel accuracy as the final quality measure, since this measure is analogous to the classification accuracy in the traditional convolutional neural networks that classify an entire image into one class. The difference is that instead of computing accuracy on an image level, it is computed on a pixel level. As can be seen from Table 1, the final accuracy for our FCN-8s deep learning model was therefore 92·18 %. Additionally, the ratios between the quality measures seem consistent with those of Long *et al.*[34].

Due to the higher accuracy, the predictions for the training subset offer more detail than those for the other two subsets and are very close to the ground truth, with the only exception being very small food items, such as onion rings, as can be seen in the training predictions image in Fig. 2. However, despite the lower amount of detail, the majority of the predictions for the other two subsets are still accurate. There are some misclassifications in the data set, such as parts of the pear and small parts of the background in the validation predictions image in Fig. 2, but these errors are rare. A more common

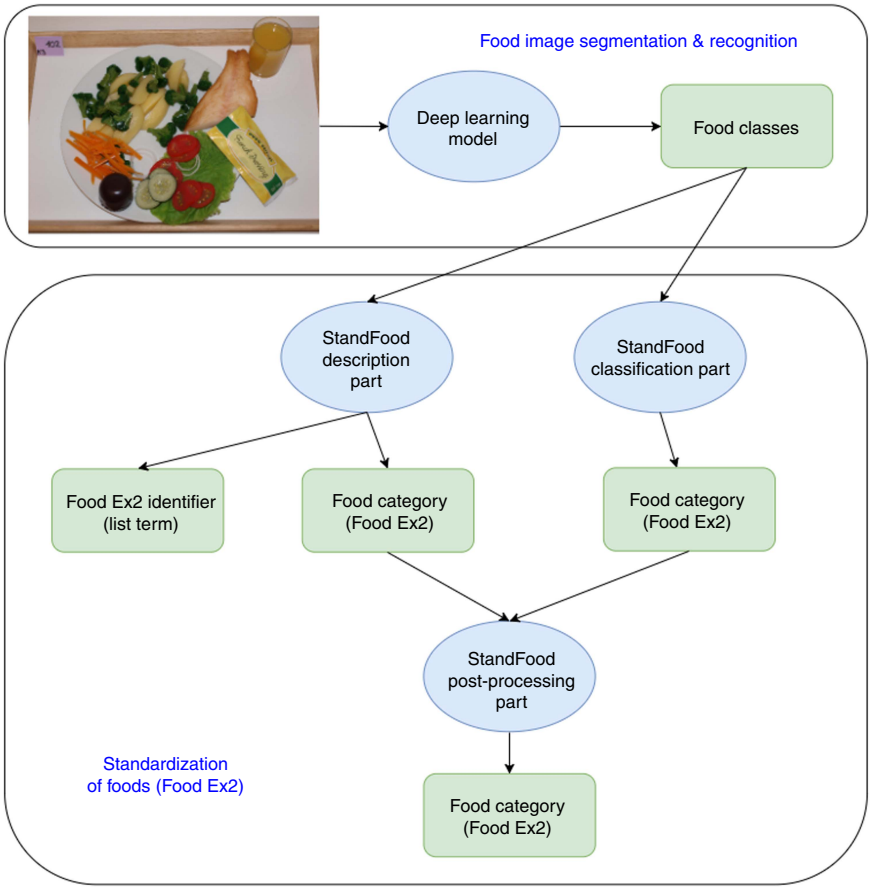

Fig. 1 Methodology flowchart. The food image recognition process uses a fake-food image to find classes (names) for all food items in the image. These are then processed by the StandFood method to define the FoodEx2 descriptors of the recognized food items. Once both the food names and descriptors are identified, the recognized fake foods can be matched with compositional data from the food composition database. The final result is a fake-food image standardized with unique descriptors, which enables food intake conversion into nutrient intake and helps the automated dietary assessment

occurrence that lowers accuracy is when the predictions do not cover the food and drink items exactly.

Results from food matching and standardization

To support the process of automated dietary assessment, each fake-food item needs to be automatically matched to nutrient data from an FCDB.

The result for each fake-food item obtained using the deep learning model is one of the fifty-five foods (food classes) for which the model is trained and is used in the FFB method. In this task we used StandFood to standardize each food class that results from the deep learning model. For this reason, we used the English names of the fifty-five food classes. First, for each food class, the classification part of StandFood is used to obtain its food category (raw, derivative, simple or aggregated composite food). The food class is also used with the description part to obtain its list term (i.e. the FoodEx2 identifier). After these two parts, their results are combined to improve the classification of the food class, in case the model used in the classification part incorrectly classifies it.

Table 2 presents the results from the StandFood classification part of four randomly selected but correctly classified food classes, one per food category. The StandFood classification part has an accuracy of 75 %. This is further improved using the StandFood post-processing part, but before we used it, the result from the description part needed to be obtained.

Concerning the second part, Table 2 provides the results from the StandFood description part of four randomly selected food classes, one for each food category. As can be seen, for the first two food classes we have perfect matches, while for the next two we have multiple choices. The multiple choices happened because of the food class description. For the last two examples provided in Table 2, the food class description is too general, so the StandFood description part suggests the most relevant matches to users. For example, for the food class 'pasta', the most relevant matches provided by StandFood are 'fresh pasta' or 'dried pasta'. To distinguish between them in the

Table 1 Results from the FCN-8s deep learning model

	Pixel accuracy (%)	Mean accuracy (%)	Mean IU (%)	Frequency-weighted IU (%)
Training	93·43	81·51	72·74	89·09
Validation	90·41	65·12	55·26	84·86
Testing	92·18	70·58	61·85	87·57
All	93·33	80·78	71·99	88·95

IU, Intersection over Union.

Fig. 2 Example images from each of the three subsets (training, validation and testing) of the fake food buffet data set, along with the corresponding ground-truth label images. The third image column contains predictions from the FCN-8s deep learning model. Each colour found in the images represents a different food or drink item; these items and their corresponding colours are listed to the right of the images

Table 2 Correctly classified food classes using the StandFood classification part and description of the food classes using the StandFood description part

Correctly classified food classes using the StandFood classification part	
Food class (result from the deep learning model)	StandFood food category (according to FoodEx2)
Broccoli	Raw (r)
Sugar	Derivative (d)
Pasta	Aggregated composite (c)
White bread	Simple composite (s)
Description of food classes using the StandFood description part	
Food class (result from the deep learning model)	StandFood relevant FoodEx2 item and its descriptor
Apple	Apples (A01DJ)
Biscuit	Biscuits (A009V)
Sugar	White sugar (A032J)
	Brown sugar (A032M)
	Flavoured sugar (A032Q)
	Sugars and similar (A0BY6)
Pasta	Fresh pasta (A007F)
	Dried pasta (A007L)

process of automated dietary assessment is a really important task because they have different nutritional profiles. It follows that the description of the food classes that is the result of the deep learning model is the key to how successful the automatic food matching will be. In the present study, we evaluated the proposed methodology using the food classes described in the FFB method. The StandFood description part has an accuracy of 86 %. In the 14 % that are not correctly described, this is caused by some culture-specific foods or food classes for which the StandFood description part could not find nouns in their description. This happened because the StandFood description part uses the extracted nouns from POS tagging for each food class, and to produce its relevant match the FoodEx2 data are searched for the names that consist of at least one of the extracted nouns. In cases when nouns are not found in a food class description, the description accuracy increases to 93 % by using the extension of the description part. Two randomly selected examples in the case of fake foods when this happened are for the food classes 'French dressing' and 'herring'. After the POS tagging, 'dressing' and 'herring' were not recognized as nouns and the StandFood description part did not provide a result. However, this was solved using the Levenshtein distance between the food class and each description presented in the FoodEx2 data. In the examples of 'French dressing' and 'herring' this returned 'salad dressing' and 'herrings'.

In addition to the FoodEx2 identifier, the StandFood description part returns the FoodEx2 food category of the most relevant match. This is further combined and used in the post-processing rules together with the food category obtained by the StandFood classification part to improve the classification accuracy. Table 3 presents the results of three randomly selected food classes after the post-processing part. After the post-processing part, the classification accuracy increases to 93 %.

Table 3 StandFood post-processing result of three randomly selected food classes

Food class (result from the deep learning model)	StandFood classification food category (according to FoodEx2)	StandFood post-processing food category (according to FoodEx2)
Muffin	Raw (r)	Aggregated composite (c)
Praline	Raw (r)	Simple composite (s)
Coffee	Derivative (d)	Simple composite (s)

In addition, if we want to link these food classes to the FCDB, we need to search the FCDB for their FoodEx2 identifiers. If the FCDB lacks the FoodEx2 identifiers, StandFood can be used to find these identifiers and to describe all the food items that exist in it.

Discussion

In the current study we have developed an advanced methodology for automatic food image recognition and the standardization of food items that supports the process of automated dietary assessment. The methodology was evaluated using food images collected using the FFB method.

Since this is the first automatic solution for recognizing the images of fake foods, we consider our results as a baseline for any future studies. Directly comparing our pixel accuracy with the classification accuracy results of other food image recognition solutions[16–27] is not appropriate because not only were those solutions tested on different data sets with a different number of food classes, but there is also a difference in the performance measures used and in the image variance; fake food generally exhibits less variance than real food, as real food can be prepared in multiple ways, which can affect its visual appearance. There have been some food recognition solutions that apply pixel-level segmentation in the past, but only one that uses deep learning[22]. However, even that one uses manually defined feature descriptors for the segmentation phase and deep learning only for the classification, so to the best of our knowledge the present study is the first that applies a single deep learning network for the joint segmentation and classification of food items. The study's results provide a base for an automated dietary assessment solution.

As the food-matching approach also enables the semi-automated assignment of food descriptors (with respect to the selected food classification and indexing system, such as FoodEx2), the linkage of food items with any FCDB complying with the selected food classification and indexing system can be performed.

Automation of the recognition of fake foods and matching them with information from a nutrient database offers great potential for research. In particular, it would reduce the effort to collect and analyse the data; that is, foods selected by participants can be assessed from photographs instead of by manual handling. In practice, the simplest approach would be to implement the solution proposed herein in a smartphone app, which would allow researchers to automatically gain relevant information about the selected foods by taking a photograph using the smartphone's camera, thus allowing them to instantaneously analyse the data. This type of automation would also reduce the biases introduced by human errors in the data and would facilitate data standardization, comparison and exchange between different laboratories using this research tool. Research questions, such as which food groups were selected more often, could be investigated automatically. The matching also allows us to study patterns in food choice (e.g. which foods are selected in combination, etc.). It can also facilitate secondary data analysis on fake-food studies, where photographs have been taken. Photographs from different experiments and laboratories could be combined for this.

Future work includes an extension of this methodology with a tool that automatically measures weight (e.g. food scape lab), or a technology that automatically estimates food volume, as this is currently the only missing part in the process of automated dietary assessment. Although the predictions from the deep learning model for the validation and testing images are not as detailed as for the training ones, they still describe the food and drink items with an accuracy that could also be sufficient for a food and drink volume estimation when paired with either a reference object or a fixed-distance camera.

Acknowledgements

Financial support: This work was supported by the project RICHFIELDS, which received funding from the European Union's Horizon 2020 research and innovation programme (grant number 654280). This work was also supported by the project ISO-FOOD, which received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration (grant agreement number 621329; 2014–2019). B.K.S. acknowledges the financial support from the Slovenian Research Agency (research core funding number P2-0098). T.B. acknowledges financial support from the School of Health Sciences and the Faculty of Health and Medicine of the University of Newcastle, Australia. The views expressed are those of the authors and not necessarily those of the funders for research, technological development and demonstration. The funders had no role in the design, analysis or writing of this article. *Conflict of interest:* The authors declare no conflict of interest. *Authorship:* B.K.S. and T.B. formulated the research question. T.B. provided experimental data about the FFB. S.M. designed and developed the food image recognition method. T.E. designed and developed the food matching and standardization method. All authors contributed to writing of the article and approved the final manuscript. *Ethics of human subject participation:* Not applicable.

References

1. Bucher T, van der Horst K & Siegrist M (2012) The fake food buffet – a new method in nutrition behaviour research. *Br J Nutr* **107**, 1553–1560.
2. Bucher T, van der Horst K & Siegrist M (2013) Fruit for dessert. How people compose healthier meals. *Appetite* **60**, 74–80.
3. Libotte E, Siegrist M & Bucher T (2014) The influence of plate size on meal composition. Literature review and experiment. *Appetite* **82**, 91–96.
4. Bucher T, Müller B & Siegrist M (2015) What is healthy food? Objective nutrient profile scores and subjective lay evaluations in comparison. *Appetite* **95**, 408–414.
5. Motteli S, Keller C, Siegrist M *et al.* (2016) Consumers' practical understanding of healthy food choices: a fake food experiment. *Br J Nutr* **116**, 559–566.
6. Nasco International, Inc. (2017) Health Education | Nutrition | Food Replicas. https://www.enasco.com/t/Health-Education/Nutrition/Food-Replicas (accessed July 2017).
7. Bucher T, Siegrist M & Van Der Horst K (2014) Vegetable variety: an effective strategy to increase vegetable choice in children. *Public Health Nutr* **17**, 1232–1236.
8. Bucher T, van der Horst K & Siegrist M (2011) Improvement of meal composition by vegetable variety. *Public Health Nutr* **14**, 1357–1363.
9. Brown HM, De Vlieger NM, Collins C *et al.* (2016) The influence of front-of-pack nutrition information on consumers' portion size perceptions. *Health Promot J Aust* **28**, 144–147.
10. De Vlieger NM, Collins C & Bucher T (2017) What is a nutritious snack? Level of processing and macronutrient content influences young adults' perceptions. *Appetite* **114**, 55–63.
11. König LM, Giese H, Schupp HT *et al.* (2016) The environment makes a difference: the impact of explicit and implicit attitudes as precursors in different food choice tasks. *Front Psychol* **7**, 1301.
12. Sproesser G, Kohlbrenner V, Schupp H *et al.* (2015) I eat healthier than you: differences in healthy and unhealthy food choices for oneself and for others. *Nutrients* **7**, 4638–4660.
13. Chen M, Dhingra K, Wu W *et al.* (2009) PFID: Pittsburgh fast-food image dataset. In *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, 7–10 November 2009, pp. 289–292. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/5413511/
14. Joutou T & Yanai K (2009) A food image recognition system with multiple kernel learning. In *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, 7–10 November 2009, pp. 285–288. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/5413400/
15. Yang S, Chen M, Pomerlau D *et al.* (2010) Food recognition using statistics of pairwise local features. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 13–18 June 2010, pp. 2249–2256. New York: Institute of Electrical and

Electronics Engineers; available at http://ieeexplore.ieee.org/document/5539907/
16. Kawano Y & Yanai K (2014) Food image recognition with deep convolutional features. In *UbiComp'14 Adjunct. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle, WA, USA, 13–17 September 2014, pp. 589–593. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?doid=2638728.2641339
17. Kagaya H, Aizawa K & Ogawa M (2014) Food detection and recognition using convolutional neural network. In *MM'14, Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL, USA, 3–7 November 2014, pp. 1055–1088. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2654970
18. Yanai K & Kawano Y (2015) Food image recognition using deep convolutional network with pre-training and fine-tuning. In *Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Turin, Italy, 29 June–3 July 2015, pp. 1–6. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/7169816/
19. Christodoulidis S, Anthimopoulos MM & Mougiakakou SG (2015) Food recognition for dietary assessment using deep convolutional neural networks. In *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops. ICIAP 2015. Lecture Notes in Computer Science*, vol. 9281, pp. 458–465 [V Murino, E Puppo, D Sona *et al.*, editors]. Cham: Springer.
20. Singla A, Yuan L & Ebrahimi T (2016) Food/non-food image classification and food categorization using pre-trained GoogLeNet model. In *MADiMa'16, Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, Amsterdam, 16 October 2016, pp. 3–11. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2986039
21. Liu C, Cao Y, Luo Y *et al.* (2016) DeepFood: deep learning-based food image recognition for computer-aided dietary assessment. In *Inclusive Smart Cities and Digital Health. ICOST 2016. Lecture Notes in Computer Science*, vol. 9677, pp. 37–48 [C Chang, L, Chiari, Y Cao *et al.*, editors]. Cham: Springer.
22. Ciocca G, Napoletano P & Schettini R (2017) Food recognition: a new dataset, experiments, and results. *IEEE J Biomed Health Inform* **21**, 588–598.
23. Hassannejad H, Matrella G, Ciampolini P *et al.* (2016) Food image recognition using very deep convolutional networks. In *MADiMa'16, Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, Amsterdam, 16 October 2016, pp. 41–49. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2986042
24. Mezgec S & Koroušić Seljak B (2017) NutriNet: a deep learning food and drink image recognition system for dietary assessment. *Nutrients* **9**, E657.
25. Chen J, Pang L & Ngo CW (2016) Deep-based ingredient recognition for cooking recipe retrieval. In *MADiMa'16, Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, Amsterdam, 16 October 2016, pp. 32–41. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2964315
26. Chen J, Pang L & Ngo CW (2017) Cross-modal recipe retrieval: how to cook this dish? In *MultiMedia Modeling. MMM 2017. Lecture Notes in Computer Science*, vol. 10132, pp. 588–600 [L Amsaleg, G Guðmundsson, C Gurrin *et al.*, editors]. Cham: Springer.
27. Salvador H, Hynes N, Aytar Y *et al.* (2017) Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017, pp. 3068–3076. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/8099810/
28. European Food Safety Authority (2017) The Food Classification and Description System FoodEx2, 2nd ed. https://www.efsa.europa.eu/en/data/data-standardisation (accessed July 2017).
29. Danish Food Informatics (2012) LanguaL™ – the International Framework for Food Description. http://www.langual.org (accessed July 2017).
30. European Food Information Resource (2009) European Food Information Resource. http://www.eurofir.org (accessed July 2017).
31. Eftimov T & Koroušić Seljak B (2015) POS tagging-probability weighted method for matching the Internet recipe ingredients with food composition data. In *Proceedings of the 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, Lisbon, 12–14 November 2015, vol. 1, pp. 330–336. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/7526937/
32. Eftimov T, Korošec P & Koroušić Seljak B (2017) StandFood: standardization of foods using a semi-automatic system for classifying and describing foods according to foodEx2. *Nutrients* **9**, E542.
33. Krizhevsky A, Sutskever I & Hinton G (2012) ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS'12, Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, NV, USA, 3–6 December 2012, vol. 1, pp. 1097–1105. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2999257
34. Long J, Shelhamer E & Darrell T (2015) Fully convolutional networks for semantic segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015, pp. 3431–3440. New York: Institute of Electrical and Electronics Engineers; available at http://ieeexplore.ieee.org/document/7298965/
35. PASCAL Visual Object Classes (2012) Homepage. http://host.robots.ox.ac.uk/pascal/VOC (accessed July 2017).
36. Jia Y, Shelhamer E, Donahue J *et al.* (2014) Caffe: convolutional architecture for fast feature embedding. In *MM'14, Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL, USA, 3–7 November 2014, pp. 675–678. New York: Association for Computing Machinery; available at https://dl.acm.org/citation.cfm?id=2654889
37. NVIDIA Corporation (1993) NVIDIA DIGITS – Interactive Deep Learning GPU Training System. https://developer.nvidia.com/digits (accessed July 2017).
38. Kingma DP & Ba J (2014) Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980 (accessed March 2018).
39. GitHub, Inc. (2008) Evaluation Metrics for Image Segmentation. http://github.com/martinkersner/py_img_seg_eval (accessed July 2017).
40. Michalski RS, Carbonell JG & Mitchell TM (editors) (2013) *Machine Learning: An Artificial Intelligence Approach*. Berlin/Heidelberg: Springer Science & Business Media.
41. Manning CD & Schütze H (1999) *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
42. Voutilainen A (2003) Part-of-speech tagging. In *The Oxford Handbook of Computational Linguistics*, pp. 219–232 [R Mitkov, editor]. Oxford: Oxford University Press.
43. Hall PA & Dowling GR (1980) Approximate string matching. *ACM Comput Surv (CSUR)* **12**, 381–402.

Chapter 4

Deep Neural Networks for Image-Based Dietary Assessment

After validating the food image segmentation approach on fake-food images, the final research step was to develop a similar solution for images of real food and test its performance on real-world food images. When considering real food, there are generally two types of food image datasets—datasets containing images, taken in controlled environments, and real-world datasets. The former are common in other research works and they include datasets that focus only on certain food types [13], datasets of images, taken in cafeterias [12], etc. These images are often idealized and a DCNN model trained on them can perform poorly on real-world images.

Real-world food and drink images can contain several additional issues that images, captured in controlled environments, generally do not, such as food occlusion, poor image quality, varying photographing distance, small food item size, etc. It is therefore important to train the model on real-world images in order to develop an accurate application for dietary assessment.

Recently, the Food Recognition Challenge [19] was held to evaluate different approaches to the recognition of multiple food and drink items per image. It introduced a large-scale dataset of real-world food images that serves as a benchmark in the field. In the scope of this research work, an approach that uses the hybrid task cascade (HTC) method [29] with a ResNet [26] backbone was developed. This approach was used on an augmented FRC dataset by training a model on it, and this model was submitted to the FRC. The HTC ResNet solution ranked second in the second round of the challenge—results are further described in the publication below. Additionally, this publication contains implementation details about the research work, presented in Chapters 2 and 3.

Permission to include the publication *"Deep neural networks for image-based dietary assessment"* [18] in this doctoral dissertation was confirmed by the *Journal of Visualized Experiments* in an email exchange from 12 June 2021.

jove

Deep Neural Networks for Image-Based Dietary Assessment

Simon Mezgec[1], **Barbara Koroušić Seljak**[2]

[1] Jožef Stefan International Postgraduate School [2] Computer Systems Department, Jožef Stefan Institute

Corresponding Author

Simon Mezgec
simon.mezgec@gmail.com

Citation

Mezgec, S., Koroušić Seljak, B. Deep Neural Networks for Image-Based Dietary Assessment. *J. Vis. Exp.* (169), e61906, doi:10.3791/61906 (2021).

Date Published

March 13, 2021

DOI

10.3791/61906

URL

jove.com/video/61906

Abstract

Due to the issues and costs associated with manual dietary assessment approaches, automated solutions are required to ease and speed up the work and increase its quality. Today, automated solutions are able to record a person's dietary intake in a much simpler way, such as by taking an image with a smartphone camera. In this article, we will focus on such image-based approaches to dietary assessment. For the food image recognition problem, deep neural networks have achieved the state of the art in recent years, and we present our work in this field. In particular, we first describe the method for food and beverage image recognition using a deep neural network architecture, called NutriNet. This method, like most research done in the early days of deep learning-based food image recognition, is limited to one output per image, and therefore unsuitable for images with multiple food or beverage items. That is why approaches that perform food image segmentation are considerably more robust, as they are able to identify any number of food or beverage items in the image. We therefore also present two methods for food image segmentation - one is based on fully convolutional networks (FCNs), and the other on deep residual networks (ResNet).

Introduction

Dietary assessment is a crucial step in determining actionable areas of an individual's diet. However, performing dietary assessment using traditionally manual approaches is associated with considerable costs. These approaches are also prone to errors as they often rely on self-reporting by the individual. Automated dietary assessment addresses these issues by providing a simpler way to quantify and qualify food intake. Such an approach can also alleviate some of the errors present in manual approaches, such as missed meals, inability to accurately assess food volume, etc. Therefore, there are clear benefits to automating dietary assessment by developing solutions that identify different foods and beverages and quantify food intake[1]. These solutions can also be used to enable an estimation of nutritional values of food and beverage items (henceforth 'food items'). Consequently, automated dietary assessment is useful for multiple applications - from strictly medical uses, such as allowing dietitians to more easily and accurately track

and analyze their patients' diets, to the usage inside well-being apps targeted at the general population.

Automatically recognizing food items from images is a challenging computer vision problem. This is due to foods being typically deformable objects, and due to the fact that a large amount of the food item's visual information can be lost during its preparation. Additionally, different foods can appear to be very similar to each other, and the same food can appear to be substantially different on multiple images[2]. Furthermore, the recognition accuracy depends on many more factors, such as image quality, whether the food item is obstructed by another item, distance from which the image was taken, etc. Recognizing beverage items presents its own set of challenges, the main one being the limited amount of visual information that is available in an image. This information could be the beverage color, beverage container color and structure, and, under optimal image conditions, the beverage density[2].

To successfully recognize food items from images, it is necessary to learn features of each food and beverage class. This was traditionally done using manually-defined feature extractors[3,4,5,6] that perform recognition based on specific item features like color, texture, size, etc., or a combination of these features. Examples of these feature extractors include multiple kernel learning[4], pairwise local features[5] and the bag-of-features model[6]. Due to the complexity of food images, these approaches mostly achieved a low classification accuracy - between 10% and 40%[3,4,5]. The reason for this is that the manual approach is not robust enough to be sufficiently accurate. Because a food item can vary significantly in appearance, it is not feasible to encompass all these variances manually. Higher classification accuracy can be achieved with manually-defined feature extractors when either the number of food classes is reduced[5], or different image features are combined[6], thus indicating that there is a need for more complex solutions to this problem.

This is why deep learning proved to be so effective for the food image recognition problem. Deep learning, or deep neural networks, was inspired by biological brains, and allows computational models composed of multiple processing layers to automatically learn features through training on a set of input images[7,8]. Because of this, deep learning has substantially improved the state of the art in a variety of research fields[7], with computer vision, and subsequently food image recognition, being one of them[2].

In particular, deep convolutional neural networks (DCNNs) are most popular for food image recognition - these networks are inspired by the visual system of animals, where individual neurons try to gain an understanding of the visual input by reacting to overlapping regions in the visual field[9]. A convolutional neural network takes the input image and performs a series of operations in each of the network layers, the most common of which are convolutional, fully-connected and pooling layers. Convolutional layers contain learnable filters that respond to certain features in the input data, whereas fully-connected layers compose output data from other layers to gain higher-level knowledge from it. The goal of pooling layers is to down-sample the input data[2]. There are two approaches to using deep learning models that proved popular: taking an existing deep neural network definition[10,11], referred to as a deep learning architecture in this article, or defining a new deep learning architecture[12,13], and training either one of these on a food image dataset. There are strengths and weaknesses to both approaches - when using an existing deep learning architecture, an

architecture that performed well for other problems can be chosen and fine-tuned for the desired problem, thus saving time and ensuring that a validated architecture has been chosen. Defining a new deep learning architecture, on the other hand, is more time-intensive, but allows the development of architectures that are specifically made to take into account the specifics of a problem and thus theoretically perform better for that problem.

In this article, we present both approaches. For the food image recognition problem, we developed a novel DCNN architecture called NutriNet[2], which is a modification of the well-known AlexNet architecture[14]. There are two main differences compared to AlexNet: NutriNet accepts 512x512-pixel images as input (as opposed to 256x256-pixel images for AlexNet), and NutriNet has an additional convolutional layer at the beginning of the neural network. These two changes were introduced in order to extract as much information from the recognition dataset images as possible. Having higher-resolution images meant that there is more information present on images and having more convolutional layers meant that additional knowledge could be extracted from the images. Compared to AlexNet's around 60 million parameters, NutriNet contains less parameters: approximately 33 million. This is because of the difference in dimensionality at the first fully-connected layer caused by the additional convolutional layer[2]. **Figure 1** contains a diagram of the NutriNet architecture. The food images that were used to train the NutriNet model were gathered from the Internet - the procedure is described in the protocol text.

For the food image segmentation problem, we used two different existing architectures: fully convolutional networks (FCNs)[15] and deep residual networks (ResNet)[16], both of which represented the state of the art for image segmentation when we used them to develop their respective food image segmentation solutions. There are multiple FCN variants that were introduced by Long et al.: FCN-32s, FCN-16s and FCN-8s[15]. FCN-32s outputs a pixel map based on the predictions by the FCN's final layer, whereas the FCN-16s variant combines these predictions with those by an earlier layer. FCN-8s considers yet another layer's predictions and is therefore able to make predictions at the finest grain, which is why it is suitable for food image recognition. The FCN-8s that we used was pre-trained on the PASCAL Visual Object Classes (PASCAL VOC) dataset[17] and trained and tested on images of food replicas (henceforth 'fake food')[18] due to their visual resemblance to real food and due to a lack of annotated images of real food on a pixel level. Fake food is used in different behavioral studies and images are taken for all dishes from all study participants. Because the food contents of these images are known, it makes the image dataset useful for deep learning model training. Dataset processing steps are described in the protocol text.

The ResNet-based solution was developed in the scope of the Food Recognition Challenge (FRC)[19]. It uses the Hybrid Task Cascade (HTC)[20] method with a ResNet-101[16] backbone. This is a state-of-the-art approach for the image segmentation problem that can use different feature extractors, or backbones. We considered other backbone networks as well, particularly other ResNet variants such as ResNet-50[16], but ResNet-101 was the most suitable due to its depth and ability to represent input images in a complex enough manner. The dataset used for training the HTC ResNet-101 model was the FRC dataset with added augmented images. These augmentations are presented in the protocol text.

This article is intended as a resource for machine learning experts looking for information about which deep learning architectures and data augmentation steps perform well for the problems of food image recognition and segmentation, as well as for nutrition researchers looking to use our approach to automate food image recognition for use in dietary assessment. In the paragraphs below, deep learning solutions and datasets from the food image recognition field are presented. In the protocol text, we detail how each of the three approaches was used to train deep neural network models that can be used for automated dietary assessment. Additionally, each protocol section contains a description of how the food image datasets used for training and testing were acquired and processed.

DCNNs generally achieved substantially better results than other methods for food image recognition and segmentation, which is why the vast majority of recent research in the field is based on these networks. Kawano et al. used DCNNs to complement manual approaches[21] and achieved a classification accuracy of 72.26% on the UEC-FOOD100 dataset[22]. Christodoulidis et al. used them exclusively to achieve a higher accuracy of 84.90% on a self-acquired dataset[23]. Tanno et al. developed DeepFoodCam - a smartphone app for food image recognition that uses DCNNs[24]. Liu et al. presented a system that performs an Internet of Things-based dietary assessment using DCNNs[25]. Martinel et al. introduced a DCNN-based approach that exploits the specifics of food images[26] and reported an accuracy of 90.27% on the Food-101 dataset[27]. Zhou et al. authored a review of deep learning solutions in the food domain[28].

Recently, Zhao et al. proposed a network specifically for food image recognition in mobile applications[29]. This approach uses a smaller 'student' network that learns from a larger 'teacher' network. With it, they managed to achieve an accuracy of 84% on the UEC-FOOD256[30] and an accuracy of 91.2% on the Food-101 dataset[27]. Hafiz et al. used DCNNs to develop a beverage-only image recognition solution and reported a very high accuracy of 98.51%[31]. Shimoda et al. described a novel method for detecting plate regions in food images without the usage of pixel-wise annotation[32]. Ciocca et al. introduced a new dataset containing food items from 20 different food classes in 11 different states (solid, sliced, creamy paste, etc.) and presented their approach for training recognition models that are able to recognize the food state, in addition to the food class[33]. Knez et al. evaluated food image recognition solutions for mobile devices[34]. Finally, Furtado et al. conducted a study on how the human visual system compares to the performance of DCNNs and found that human recognition still outperforms DCNNs with an accuracy of 80% versus 74.5%[35]. The authors noted that with a small number of food classes, the DCNNs perform well, but on a dataset with hundreds of classes, human recognition accuracy is higher[35], highlighting the complexity of the problem.

Despite its state-of-the-art results, deep learning has a major drawback - it requires a large input dataset to train the model on. In the case of food image recognition, a large food image dataset is required, and this dataset needs to encompass as many different real-world scenarios as possible. In practice this means that for each individual food or beverage item, a large collection of images is required, and as many different items as possible need to be present in the dataset. If there are not enough images for a specific item in the dataset, that item is unlikely to be recognized successfully. On the other hand, if only a small number of items is covered by the

dataset, the solution will be limited in scope, and only able to recognize a handful of different foods and beverages.

Multiple datasets were made available in the past. The Pittsburgh Fast-Food Image Dataset (PFID)[3] was introduced to encourage more research in the field of food image recognition. The University of Electro-Communications Food 100 (UEC-FOOD100)[22] and University of Electro-Communications Food 256 (UEC-FOOD256)[30] datasets contain Japanese dishes, expanded with some international dishes in the case of the UEC-FOOD256 dataset. The Food-101 dataset contains popular dishes acquired from a website[27]. The Food-50[36] and Video Retrieval Group Food 172 (VireoFood-172)[37] datasets are Chinese-based collections of food images. The University of Milano-Bicocca 2016 (UNIMIB2016) dataset is composed of images of food trays from an Italian canteen[38]. Recipe1M is a large-scale dataset of cooking recipes and food images[39]. The Food-475 dataset[40] collects four previously published food image datasets[27,30,36,37] into one. The Beijing Technology and Business University Food 60 (BTBUFood-60) is a dataset of images meant for food detection[41]. Recently, the ISIA Food-500 dataset[42] of miscellaneous food images was made available. In comparison to other publicly available food image datasets, it contains a large number of images, divided into 500 food classes, and is meant to advance the development of multimedia food recognition solutions[42].

Protocol

1. Food image recognition with NutriNet

1. **Obtaining the food image dataset**

    1. Gather a list of different foods and beverages that will be the outputs of the food image recognition model. A varied list of popular foods and beverages is preferred, as that will allow the training of a robust food image recognition model.

    2. Save the food and beverage list in a text file (e.g., 'txt' or 'csv').
    NOTE: The text file used by the authors of this article can be found in the supplemental files ('food_items.txt') and includes a list of 520 Slovenian food items.

    3. Write or download a Python[43] script that uses the Google Custom Search API[44] to download images of each food item from the list and saves them into a separate folder for each food item.
    NOTE: The Python script used by the authors of this article can be found in the supplemental files ('download_images.py'). If this script is used, the Developer Key (variable 'developerKey', line 8 in the Python script code) and Custom Search Engine ID (variable 'cx', line 28 in the Python script code) need to be replaced with values specific to the Google account being used.

    4. Run the Python script from step 1.1.3 (e.g., with the command: 'python download_images.py').

2. **(Optional) Cleaning the food image dataset**

    1. Train a food image detection model in the same way as in section 1.4, except use only two outputs (food, non-food) as opposed to the list of outputs from step 1.1.1.
    NOTE: The authors of this article used images combined from recipe websites and the ImageNet dataset[45] to train the food image detection model. Since the focus here is on food image recognition and this is an optional step for cleaning the recognition dataset, further details are omitted.

Instead, more details about this approach can be found in Mezgec et al.[2].

2. Run the detection model from step 1.2.1 on the food image dataset that is the result of step 1.1.4.

3. Delete every image that was tagged as non-food by the detection model from step 1.2.1.

4. Manually check the food image dataset for other erroneous or low-quality images, and for image duplicates.

5. Delete images found in step 1.2.4.

3. **Augmenting the food image dataset**

1. Create a new version of each image from the food image dataset by rotating it by 90° using the CLoDSA library[46] (lines 19 to 21 in the included Python script).

NOTE: The Python script containing all the CLoDSA commands used by the authors of this article can be found in a file included in the supplemental files ('nutrinet_augmentation.py'). If this script is used, the Input Path (variable 'INPUT_PATH', line 8 in the Python script code) and Output Path (variable 'OUTPUT_PATH', line 11 in the Python script code) need to be replaced with paths to the desired folders.

2. Create a new version of each image from the food image dataset by rotating it by 180° using the CLoDSA library (lines 19 to 21 in the included Python script).

3. Create a new version of each image from the food image dataset by rotating it by 270° using the CLoDSA library (lines 19 to 21 in the included Python script).

4. Create a new version of each image from the food image dataset by flipping it horizontally using the CLoDSA library (lines 23 and 24 in the included Python script).

5. Create a new version of each image from the food image dataset by adding random color noise to it using the CLoDSA library (lines 26 and 27 in the included Python script).

6. Create a new version of each image from the food image dataset by zooming into it by 25% using the CLoDSA library (lines 29 and 30 in the included Python script).

7. Save images from steps 1.3.1-1.3.6, along with the original images (lines 16 and 17 in the included Python script), into a new food image dataset (in total, 7 variants per food image). This is done by executing the command in line 32 of the included Python script.

4. **Performing food image recognition**

1. Import the food image dataset from step 1.3.7 into the NVIDIA DIGITS environment[47], dividing the dataset into training, validation and testing subsets in the NVIDIA DIGITS user interface.

2. Copy and paste the definition text of the NutriNet architecture[2] into NVIDIA DIGITS as a custom network.

NOTE: The NutriNet architecture definition text can be found in the supplemental files ('nutrinet.prototxt').

3. (Optional) Define training hyperparameters in the NVIDIA DIGITS user interface.

NOTE: Hyperparameters are parameters that are used to define the training process prior to its start. The hyperparameters used by the authors of this article can be found in a file included in the supplemental files ('nutrinet_hyperparameters.prototxt'). While experimentation is needed for each dataset to find the optimal hyperparameters, the file contains a hyperparameter configuration which can be copied into the NVIDIA DIGITS user interface. Furthermore, NVIDIA DIGITS populates the hyperparameters with default values which can be used as a baseline. This step is therefore optional.

4. Run the training of the NutriNet model.

5. After training is complete, take the best-performing NutriNet model iteration. This model is then used for testing the performance of this approach.

   NOTE: There are multiple ways to determine the best-performing model iteration. A straightforward way to do this is as follows. NVIDIA DIGITS outputs a graph of accuracy measures for each training epoch. Check which epoch achieved the lowest loss value for the validation subset of the food image dataset - that model iteration can be considered best-performing. An optional step in determining the best-performing model iteration is to observe how the loss value for the training subset changes from epoch to epoch and if it starts to drop continuously while the loss value for the validation subset remains the same or rises continuously, take the epoch prior to this drop in training loss value, as that can signal when the model started overfitting on the training images.

2. Food image segmentation with FCNs

1. **Obtaining the fake-food image dataset**

   1. Obtain a dataset of fake-food images. Fake-food images are gathered by researchers conducting behavioral studies using food replicas.

      NOTE: The authors of this article received images of fake food that were collected in a lab environment[18].

   2. Manually annotate every food image on a pixel level - each pixel in the image must contain information about which food class it belongs to. The result of this step is one annotation image for each image from the food image dataset, where each pixel represents one of the food classes.

      NOTE: There are many tools to achieve this - the authors of this article used JavaScript Segment Annotator[48].

2. **Augmenting the fake-food image dataset**

   1. Perform the same steps as in section 1.3, but only on images from the training subset of the food image dataset.

      NOTE: With the exception of step 1.3.5, all data augmentation steps need to be performed on corresponding annotation images as well. If the script from section 1.3 is used, the Input Path (variable 'INPUT_PATH', line 8 in the Python[43] script code) and Output Path (variable 'OUTPUT_PATH', line 11 in the Python script code) need to be replaced with paths to the desired folders. In addition, set the Problem (variable 'PROBLEM', line 6 in the Python script code) to 'instance_segmentation' and the Annotation Mode (variable 'ANNOTATION_MODE', line 7 in the

Python script code) and Output Mode (variable 'OUTPUT_MODE', line 10 in the Python script code) to 'coco'.

3. **Performing fake-food image segmentation**

    1. Perform the same steps as in section 1.4, with the exception of step 1.4.2. In place of that step, perform steps 2.3.2 and 2.3.3.

        NOTE: Hyperparameters are parameters that are used to define the training process prior to its start. The training hyperparameters used by the authors of this article for the optional step 1.4.3 can be found in a file included in the supplemental files ('fcn-8s_hyperparameters.prototxt'). While experimentation is needed for each dataset to find the optimal set of hyperparameters, the file contains a hyperparameter configuration which can be copied into the NVIDIA DIGITS[47] user interface. Furthermore, NVIDIA DIGITS populates the hyperparameters with default values which can be used as a baseline.

    2. Copy and paste the definition text of the FCN-8s architecture[15] into the NVIDIA DIGITS environment as a custom network.

        NOTE: The FCN-8s architecture definition text is publicly available on GitHub[49].

    3. Enter the path to the pre-trained FCN-8s model weights into the NVIDIA DIGITS user interface.

        NOTE: These model weights were pre-trained on the PASCAL VOC dataset[17] and can be found on the Internet[49].

3. Food image segmentation with HTC ResNet

1. **Obtaining the food image dataset**

    1. Download the food image dataset from the FRC website[19].

2. **Augmenting the food image dataset**

    1. Perform steps 1.3.1-1.3.4.

        NOTE: The Python[43] script containing all the CLoDSA[46] commands used by the authors of this article can be found in a file included in the supplemental files ('frc_augmentation.py'). If this script is used, the Input Path (variable 'INPUT_PATH', line 8 in the Python script code) and Output Path (variable 'OUTPUT_PATH', line 11 in the Python script code) need to be replaced with paths to the desired folders.

    2. Create a new version of each image from the food image dataset by adding Gaussian blur to it using the CLoDSA library (lines 26 and 27 in the included Python script).

    3. Create a new version of each image from the food image dataset by sharpening it using the CLoDSA library (lines 29 and 30 in the included Python script).

    4. Create a new version of each image from the food image dataset by applying gamma correction to it using the CLoDSA library (lines 32 and 33 in the included Python script).

    5. Save images from steps 3.2.1-3.2.4, along with the original images (lines 16 and 17 in the included Python script), into a new food image dataset (in total, 8 variants per food image). This is done by executing the command in line 35 of the included Python script.

    6. Save images from steps 3.2.2-3.2.4, along with the original images (lines 16 and 17 in the included

Python script), into a new food image dataset (in total, 4 variants per food image). This is done by deleting lines 19 to 24 of the included Python script and executing the command in line 35.

3. **Performing food image segmentation**

    1. Modify the existing HTC[20] ResNet-101 architecture[16] definition from the MMDetection library[50] in sections 'model settings' and 'dataset settings' of the architecture definition file so that it accepts the food image datasets from steps 3.1.1, 3.2.5 and 3.2.6.

    2. (Optional) Modify the HTC ResNet-101 architecture definition from step 3.3.1 to define training hyperparameters: batch size in section 'dataset settings', solver type and learning rate in section 'optimizer', learning policy in section 'learning policy' and number of training epochs in section 'runtime settings' of the architecture definition file.
    NOTE: The modified HTC ResNet-101 architecture definition file can be found in the supplemental files ('htc_resnet-101.py'). Hyperparameters are parameters that are used to define the training process prior to its start. While experimentation is needed for each dataset to find the optimal set of hyperparameters, the file already contains a hyperparameter configuration which can be used without modification. This step is therefore optional.

    3. Run the training of the HTC ResNet-101 model on the food image dataset from step 3.1.1 using the MMDetection library (e.g., with the command: 'python mmdetection/tools/train.py htc_resnet-101.py').

    4. After the training from step 3.3.3 is complete, take the best-performing HTC ResNet-101 model iteration and fine-tune it by running the next phase of training on the food image dataset from step 3.2.5.
    NOTE: There are multiple ways to determine the best-performing model iteration. A straightforward way to do this is as follows. The MMDetection library outputs values of accuracy measures for each training epoch in the command line interface. Check which epoch achieved the lowest loss value for the validation subset of the food image dataset - that model iteration can be considered best-performing. An optional step in determining the best-performing model iteration is to observe how the loss value for the training subset changes from epoch to epoch and if it starts to drop continuously while the loss value for the validation subset remains the same or rises continuously, take the epoch prior to this drop in training loss value, as that can signal when the model started overfitting on the training images.

    5. After the training from step 3.3.4 is complete, take the best-performing HTC ResNet-101 model iteration and fine-tune it by running the next phase of training on the food image dataset from step 3.2.6.
    NOTE: See note for step 3.3.4.

    6. After the training from step 3.3.5 is complete, take the best-performing HTC ResNet-101 model iteration and fine-tune it by again running the next phase of training on the food image dataset from step 3.2.5.
    NOTE: See note for step 3.3.4.

    7. After the training from step 3.3.6 is complete, take the best-performing HTC ResNet-101 model

iteration. This model is then used for testing the performance of this approach.

NOTE: See note for step 3.3.4. Steps 3.3.3-3.3.7 yielded the best results for the purposes defined by the authors of this article. Experimentation is needed for each dataset to find the optimal sequence of training and data augmentation steps.

Representative Results

NutriNet was tested against three popular deep learning architectures of the time: AlexNet[14], GoogLeNet[51] and ResNet[16]. Multiple training parameters were also tested for all architectures to define the optimal values[2]. Among these is the choice of solver type, which determines how the loss function is minimized. This function is the primary quality measure for training neural networks as it is better suited for optimization during training than classification accuracy. We tested three solvers: Stochastic Gradient Descent (SGD)[52], Nesterov's Accelerated Gradient (NAG)[53] and the Adaptive Gradient algorithm (AdaGrad)[54]. The second parameter is batch size, which defines the number of images that are processed at the same time. The depth of the deep learning architecture determined the value of this parameter, as deeper architectures require more space in the GPU memory - the consequence of this approach was that the memory was completely filled with images for all architectures, regardless of depth. The third parameter is learning rate, which defines the speed with which the neural network parameters are being changed during training. This parameter was set in unison with the batch size, as the number of concurrently processed images dictates the convergence rate. AlexNet models were trained using a batch size of 256 images and a base learning rate of 0.02; NutriNet used a batch size of 128 images and a rate of 0.01; GoogLeNet 64 images and a rate of 0.005; and ResNet 16 images and a rate of 0.00125. Three other parameters were fixed for all architectures: learning rate policy (step-down), step size (30%) and gamma (0.1). These parameters jointly describe how the learning rate is changing in every epoch. The idea behind this approach is that the learning rate is being gradually lowered to fine-tune the model the closer it gets to the optimal loss value. Finally, the number of training epochs was also fixed to 150 for all deep learning architectures[2].

The best result among all the parameters tested that NutriNet achieved was a classification accuracy of 86.72% on the recognition dataset, which was around 2% higher than the best result for AlexNet and slightly higher than GoogLeNet's best result. The best-performing architecture overall was ResNet (by around 1%), however the training time for ResNet is substantially higher compared to NutriNet (by a factor of approximately five), which is important if models are continuously re-trained to improve accuracy and the number of recognizable food items. NutriNet, AlexNet and GoogLeNet achieved their best results using the AdaGrad solver, whereas ResNet's best model used the NAG solver. NutriNet was also tested on the publicly available UNIMIB2016 food image dataset[38]. This dataset contains 3,616 images of 73 different food items. NutriNet achieved a recognition accuracy of 86.39% on this dataset, slightly outperforming the baseline recognition result of the authors of the dataset, which was 85.80%. Additionally, NutriNet was tested on a small dataset of 200 real-world images of 115 different food and beverage items, where NutriNet achieved a top-5 accuracy of 55%.

To train the FCN-8s fake-food image segmentation model, we used Adam[55] as the solver type, as we found that it performed optimally for this task. The base learning rate was set very low - to 0.0001. The reason for the low number is the fact

that only one image could be processed at a time, which is a consequence of the pixel-level classification process. The GPU memory requirements for this approach are significantly greater than image-level classification. The learning rate thus had to be set low so that the parameters were not being changed too fast and converge to less optimal values. The number of training epochs was set to 100, while the learning rate policy, step size and gamma were set to step-down, 34% and 0.1, respectively, as these parameters produced the most accurate models.

Accuracy measurements of the FCN-8s model were performed using the pixel accuracy measure[15], which is analogous to the classification accuracy of traditional deep learning networks, the main difference being that the accuracy is computed on the pixel level instead of on the image level:

$$PA = \frac{\sum_i n_{ii}}{\sum_i t_i}$$

where *PA* is the pixel accuracy measure, n_{ij} is the number of pixels from class *i* predicted to belong to class *j* and $t_i = \Sigma_j n_{ij}$ is the total number of pixels from class *i* in the ground-truth labels[1]. In other words, the pixel accuracy measure is computed by dividing correctly predicted pixels by the total number of pixels. The final accuracy of the trained FCN-8s model was 92.18%. **Figure 2** shows three example images from the fake-food image dataset (one from each of the training, validation and testing subsets), along with the corresponding ground-truth and model prediction labels.

The parameters to train the HTC[20] ResNet-101 model for food image segmentation were set as follows: the solver type used was SGD because it outperformed other solver types. The base learning rate was set to 0.00125 and the batch size to 2 images. The number of training epochs was set to 40 per training phase, and multiple training phases were performed - first on the original FRC dataset without augmented images, then on the 8x-augmented and 4x-augmented FRC dataset multiple times in an alternating fashion, each time taking the best-performing model and fine-tuning it in the next training phase. More details on the training phases can be found in section 3.3 of the protocol text. Finally, the step-down learning policy was used, with fixed epochs for when the learning rate decreased (epochs 28 and 35 for the first training phase). An important thing to note is that while this sequence of training phases produced the best results in our testing in the scope of the FRC, using another dataset might require a different sequence to produce optimal results.

This ResNet-based solution for food image segmentation was evaluated using the following precision measure[19]:

$$P_{IoU \geq 0.5} = \frac{TP_{IoU \geq 0.5}}{TP_{IoU \geq 0.5} + FP_{IoU \geq 0.5}}$$

where *P* is precision, *TP* is the number of true positive predictions by the food image segmentation model, *FP* is the number of false positive predictions and *IoU* is Intersection over Union, which is computed with this equation:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

where *Area of Overlap* represents the number of predictions by the model that overlap with the ground truth, and *Area of Union* represents the total number of predictions by the model together with the ground truth, both on a pixel level and for each individual food class. Recall is used as a secondary measure and is calculated in a similar way, using the following formula[19]:

$$R_{IoU \geq 0.5} = \frac{TP_{IoU \geq 0.5}}{TP_{IoU \geq 0.5} + FN_{IoU \geq 0.5}}$$

where *R* is recall and *FN* is the number of false negative predictions by the food image segmentation model. The precision and recall measures are then averaged across all classes in the ground truth. Using these measures, our model achieved an average precision of 59.2% and an average recall of 82.1%, which ranked second in the second round of the Food Recognition Challenge[19]. This result was 4.2% behind the first place and 5.3% ahead of the third place in terms of the average precision measure. **Table 1** contains the results for the top-4 participants in the competition.

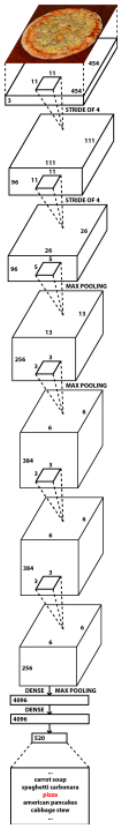

Figure 1: Diagram of the NutriNet deep neural network architecture. This figure has been published in Mezgec et al.[2].

Please click here to view a larger version of this figure.

Figure 2: Images from the fake-food image dataset. Original images (left), manually-labelled ground-truth labels (middle) and predictions from the FCN-8s model (right). This figure has been published in Mezgec et al.[1]. Please click here to view a larger version of this figure.

Team Name	Placement	Average Precision	Average Recall
rssfete	1	63.4%	88.6%
simon_mezgec	2	59.2%	82.1%
arimboux	3	53.9%	73.5%
latentvec	4	48.7%	71.1%

Table 1: Top-4 results from the second round of the Food Recognition Challenge. Average precision is taken as the primary performance measure and average recall as a secondary measure. Results are taken from the official competition leaderboard[19].

Supplemental Files. Please click here to download this File.

Discussion

In recent years, deep neural networks have been validated multiple times as a suitable solution for recognizing food images[10,11,12,21,23,25,26,29,31,33]. Our work presented in this article serves to further prove this[1,2]. The single-output food image recognition approach is straightforward and can be used for simple applications where images with only one food or beverage item are expected[2].

The food image segmentation approach seems particularly suitable for recognizing food images in general, without any restriction on the number of food items[1]. Because it works by classifying each individual pixel of the image, it is able to not only recognize any number of food items in the image, but also specify where a food item is located, as well as how large it is. The latter can then be used to perform food weight estimation, particularly if used with either a reference object or a fixed-distance camera.

There has been some work done regarding the availability of food image datasets[3,22,27,30,36,37,38,39,40,41,42], and we hope more will be done in the future, particularly when it comes to aggregating food image datasets from different regions across the world, which would enable more robust solutions to be developed. Currently, the accuracy of automatic food image recognition solutions has not yet reached human-level accuracy[35], and this is likely in large part due to the usage of food image datasets of insufficient size and quality.

In the future, our goal will be to further evaluate the developed procedures on real-world images. In general, datasets in this field often contain images taken in controlled environments or images that were manually optimized for recognition. This is why it is important to gather a large and diverse real-world food image dataset to encompass all the different food and beverage items that individuals might want to recognize. The first step towards this was provided by the Food Recognition Challenge, which included a dataset of real-world food images[19], but further work needs to be done to validate this approach on food images from all around the world and in cooperation with dietitians.

Disclosures

The authors have nothing to disclose.

Acknowledgments

The authors would like to thank Tamara Bucher from the University of Newcastle, Australia, for providing the fake-food image dataset. This work was supported by the European Union's Horizon 2020 research and innovation programs (grant numbers 863059 - FNS-Cloud, 769661 - SAAM); and the Slovenian Research Agency (grant number P2-0098). The European Union and Slovenian Research Agency had no role in the design, analysis or writing of this article.

References

1. Mezgec, S., Eftimov, T., Bucher, T., Koroušić Seljak, B. Mixed Deep Learning and Natural Language Processing Method for Fake-Food Image Recognition and Standardization to Help Automated Dietary Assessment. *Public Health Nutrition.* **22** (7), 1193-1202 (2019).
2. Mezgec, S., Koroušić Seljak, B. NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment. *Nutrients.* **9** (7), 657 (2017).
3. Chen, M. et al. PFID: Pittsburgh Fast-Food Image Dataset. *Proceedings of the ICIP 2009.* 289-292 (2009).
4. Joutou, T., Yanai, K. A Food Image Recognition System with Multiple Kernel Learning. *Proceedings of the ICIP 2009.* 285-288 (2009).
5. Yang, S., Chen, M., Pomerlau, D., Sukthankar, R. Food Recognition using Statistics of Pairwise Local Features. *Proceedings of the CVPR 2010.* 2249-2256 (2010).

6. Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., Mougiakakou, S. G. A Food Recognition System for Diabetic Patients Based on an Optimized Bag-of-Features Model. *IEEE Journal of Biomedical and Health Informatics.* **18** (4), 1261-1271 (2014).
7. LeCun, Y., Bengio, Y., Hinton, G. Deep Learning. *Nature.* **521**, 436-444 (2015).
8. Deng, L., Yu, D. Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing.* **7** (3-4), 197-387 (2014).
9. Hubel, D. H., Wiesel, T. N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology.* **160** (1), 106-154 (1962).
10. Singla, A., Yuan, L., Ebrahimi, T. Food/Non-Food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. *Proceedings of the MADiMa'16.* 3-11 (2016).
11. Yanai, K., Kawano, Y. Food Image Recognition using Deep Convolutional Network with Pre-Training and Fine-Tuning. *Proceedings of the ICMEW 2015.* 1-6 (2015).
12. Liu, C. et al. DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. *Proceedings of the ICOST 2016.* 37-48 (2016).
13. De Sousa Ribeiro, F. et al. An End-to-End Deep Neural Architecture for Optical Character Verification and Recognition in Retail Food Packaging. *Proceedings of the ICIP 2018.* 2376-2380 (2018).
14. Krizhevsky, A., Sutskever, I., Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the NIPS'12.* 1097-1105 (2012).
15. Long, J., Shelhamer, E., Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *Proceedings of the CVPR 2015.* 3431-3440 (2015).
16. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. *Proceedings of the CVPR 2016.* 770-778 (2016).
17. PASCAL VOC Project. *PASCAL Visual Object Classes.* http://host.robots.ox.ac.uk/pascal/VOC (2020).
18. Bucher, T., van der Horst, K., Siegrist, M. Fruit for Dessert. How People Compose Healthier Meals. *Appetite.* **60** (1), 74-80 (2013).
19. AICrowd. *Food Recognition Challenge.* https://www.aicrowd.com/challenges/food-recognition-challenge (2020).
20. Chen, K. et al. Hybrid Task Cascade for Instance Segmentation. *Proceedings of the CVPR 2019.* 4974-4983 (2019).
21. Kawano, Y., Yanai, K. Food Image Recognition with Deep Convolutional Features. *Proceedings of the UbiComp 2014.* 589-593 (2014).
22. Matsuda, Y., Hoashi, H., Yanai, K. Recognition of Multiple-Food Images by Detecting Candidate Regions. *Proceedings of the ICME 2012.* 25-30 (2012).
23. Christodoulidis, S., Anthimopoulos, M. M., Mougiakakou, S. G. Food Recognition for Dietary Assessment using Deep Convolutional Neural Networks. *Proceedings of the ICIAP 2015.* 458-465 (2015).
24. Tanno, R., Okamoto, K., Yanai, K. DeepFoodCam: A DCNN-Based Real-Time Mobile Food Recognition System. *Proceedings of the MADiMa'16.* 89-89 (2016).
25. Liu, C. et al. A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge

Computing Service Infrastructure. *IEEE Transactions on Services Computing.* **11** (2), 249-261 (2017).

26. Martinel, N., Foresti, G. L., Micheloni, C. Wide-Slice Residual Networks for Food Recognition. *Proceedings of the IEEE WACV 2018.* 567-576 (2018).
27. Bossard, L., Guillaumin, M., Van Gool, L. Food-101-Mining Discriminative Components with Random Forests. *Proceedings of the ECCV'14.* 446-461 (2014).
28. Zhou, L., Zhang, C., Liu, F., Qiu, Z., He, Y. Application of Deep Learning in Food: A Review. *Comprehensive Reviews in Food Science and Food Safety.* **18**, 1793-1811 (2019).
29. Zhao, H., Yap, K.-H., Kot, A. C., Duan, L. JDNet: A Joint-Learning Distilled Network for Mobile Visual Food Recognition. *IEEE Journal of Selected Topics in Signal Processing.* **14** (4), 665-675 (2020).
30. Kawano, Y., Yanai, K. Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation. *Proceedings of the ECCV'14.* 3-17 (2014).
31. Hafiz, R., Haque, M. R., Rakshit, A., Uddin, M. S. Image-Based Soft Drink Type Classification and Dietary Assessment System using Deep Convolutional Neural Network with Transfer Learning. *Journal of King Saud University - Computer and Information Sciences.* In Press (2020).
32. Shimoda, W., Yanai, K. Weakly-Supervised Plate and Food Region Segmentation. *Proceedings of the ICME 2020.* 1-6 (2020).
33. Ciocca, G., Micali, G., Napoletano, P. State Recognition of Food Images using Deep Features. *IEEE Access.* **8**, 32003-32017 (2020).
34. Knez, S., Šajn, L. Food Object Recognition using a Mobile Device: Evaluation of Currently Implemented Systems. *Trends in Food Science & Technology.* **99**, 460-471 (2020).
35. Furtado, P., Caldeira, M., Martins, P. Human Visual System vs Convolution Neural Networks in Food Recognition Task: An Empirical Comparison. *Computer Vision and Image Understanding.* **191**, 102878 (2020).
36. Chen, M.-Y. et al. Automatic Chinese Food Identification and Quantity Estimation. *SA'12 Technical Briefs.* 1-4 (2012).
37. Chen, J., Ngo, C.-W. Deep-Based Ingredient Recognition for Cooking Recipe Retrieval. *Proceedings of the MM'16.* 32-41 (2016).
38. Ciocca, G., Napoletano, P., Schettini, R. Food Recognition: A New Dataset, Experiments, and Results. *IEEE Journal of Biomedical and Health Informatics.* **21** (3), 588-598 (2017).
39. Salvador, A. et al. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *Proceedings of the IEEE CVPR 2017.* 3020-3028 (2017).
40. Ciocca, G., Napoletano, P., Schettini, R. CNN-Based Features for Retrieval and Classification of Food Images. *Computer Vision and Image Understanding.* **176 - 177**, 70-77 (2018).
41. Cai, Q., Li, J., Li, H., Weng, Y. BTBUFood-60: Dataset for Object Detection in Food Field. *Proceedings of the IEEE BigComp 2019.* 1-4 (2019).
42. Min, W. et al. ISIA Food-500: A Dataset for Large-Scale Food Recognition via Stacked Global-Local Attention Network. *Proceedings of the MM'20.* 393-401 (2020).

43. Python Software Foundation. *Python.* https://www.python.org (2020).
44. Google. *Google Custom Search API.* https://developers.google.com/resources/api-libraries/documentation/customsearch/v1/python/latest/customsearch_v1.cse.html (2020).
45. Stanford Vision Lab. *ImageNet.* http://www.image-net.org (2020).
46. Heras, J. *CLoDSA.* https://github.com/joheras/CLoDSA (2020).
47. NVIDIA. *NVIDIA DIGITS.* https://developer.nvidia.com/digits (2020).
48. Yamaguchi, K. *JavaScript Segment Annotator.* https://github.com/kyamagu/js-segment-annotator (2020).
49. Shelhamer, E. *Fully Convolutional Networks for Semantic Segmentation.* https://github.com/shelhamer/fcn.berkeleyvision.org (2020).
50. Multimedia Laboratory, CUHK. *MMDetection.* https://github.com/open-mmlab/mmdetection (2020).
51. Szegedy, C. et al. Going Deeper with Convolutions. *Proceedings of the CVPR 2015.* 1-9 (2015).
52. Bottou, L. Large-Scale Machine Learning with Stochastic Gradient Descent. *Proceedings of the COMPSTAT'2010.* 177-186 (2010).
53. Nesterov, Y. A Method of Solving a Convex Programming Problem with Convergence Rate O(1/k^2). *Doklady Akademii Nauk SSSR.* **27**, 372-376 (1983).
54. Duchi, J., Hazan, E., Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research.* **12**, 2121-2159 (2011).
55. Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization. *arXiv Preprint.* arXiv:1412.6980 (2017).

Chapter 5

Discussion

The research work, presented in this doctoral dissertation, started with the two hypotheses, listed in Chapter 1. The goal of the first hypothesis was to define a novel DCNN architecture that would be more accurate for the task of food and drink image recognition. The NutriNet architecture was designed as a modification of the AlexNet architecture in the scope of the research work, described in Chapter 2 [4]. The latter was modified by adding a convolutional layer to the beginning of the neural network and by increasing the resolution of the input images to 512 × 512 pixels. With that, more information could be gathered from food images, which resulted in a higher classification accuracy compared to AlexNet. This is likely a consequence of the visual complexity of food and drink items, which require more information to differentiate between different items. Because of the difference in the number of parameters between AlexNet and NutriNet, NutriNet is also considerably faster to train.

Testing results showed that NutriNet achieved the second-best classification accuracy on the self-acquired dataset. The only architecture that slightly outperformed it was ResNet. However, the training time also needs to be considered, and in that regard, NutriNet outperformed all other tested DCNN architectures when employing equal resolution of input images for all architectures—GoogLeNet and ResNet by a factor of five, and AlexNet by a factor of around 1.5. The difference in training time is substantial, and consequently NutriNet achieved the best ratio of classification accuracy to training time. NutriNet thus provides a viable alternative to the other tested architectures for food image recognition applications where the DCNN model is regularly retrained with new images.

Additionally, NutriNet outperformed the baseline result on the UNIMIB2016 food image dataset [12]. To the best of the author's knowledge, the solution for food image recognition that uses NutriNet was the first to recognize drinks in addition to food items, and the 520 different food and drink items that make up the training dataset was significantly higher than contemporary approaches [12], [30], [31]. Due to the encouraging performance of the NutriNet architecture, the first hypothesis is concluded with the finding that the development of a new DCNN architecture for food and drink image recognition that is more accurate than the architecture it is based on is both possible and reasonable.

The second hypothesis, defined in Chapter 1, states that a DCNN architecture can be used to perform the joint segmentation and classification of food and drink images. This was confirmed by the development of a solution that uses FCNs to segment and classify fake-food images [17], which is described in Chapter 3. Due to its high accuracy, the FCN-based approach can be used by researchers who employ food replicas to more quickly identify the food and drink items that participants choose in behavioral studies. In turn, this has the potential to shorten the time needed to analyze the results of such studies and accelerate progress in the field.

To the best of the author's knowledge, this solution was the first for automated food replica recognition, making it a potential benchmark for future research work in the field. Comparison with real-food recognition systems is not appropriate due to the difference in food item variance—real food exhibits a far greater difference in visual appearance from one image to the next, whereas participants in fake-food studies generally choose from an array of items that do not vary in appearance. Additionally, due to the design of FCNs and to the best of the author's knowledge, the approach was the first to jointly perform the segmentation and classification of food images in a single DCNN architecture.

As part of the Food Recognition Challenge [19], a real-food image segmentation and classification approach was developed [18]. This approach is presented in Chapter 4, and it can be considered an upgrade of the fake-food approach due to the increased complexity of recognizing real food as opposed to fake food, therefore serving to further confirm the second hypothesis. The solution is based on the HTC ResNet method, and it achieved second place in the second round of the challenge based on the precision measure, defined in the challenge. The result was 4.2% behind the first-place team and 5.3% ahead of the third-place team [19]. In total, 142 submissions were evaluated in the FRC judging system as part of this research work, and the largest increase in precision was achieved by using extensive data augmentation to generate additional image variants that could potentially appear in the real world. The challenge was organized in such a way that the evaluation of submissions was performed on a hidden testing subset of the FRC dataset [19] in order to limit the possibility of reverse-engineering a solution.

Using the FRC solution, a mobile application was developed to facilitate dietary assessment. The application is called Vid (Slovene for "vision"), and it was developed for the Android mobile operating system [32]. It works in the following way: first, the user takes a photograph with their camera application of choice. Then, the application performs the segmentation and classification of all food and drink items present in the image. Once this process is finished, the segmented image and the names of the recognized food and drink items are presented to the user, and they are saved into their food diary, along with the corresponding time and date. The food image recognition model that is used in Vid was taken from the FRC submission, which means it was trained on an augmented version of the official food image dataset of the second round of the FRC [19]. Figure 5.1 contains screenshots of Vid recognizing food items in an image.

Because Vid has no limitations regarding the number of food and drink items in any given image, it is considerably more robust and requires much less effort on the part of the user than the mobile application, developed using the NutriNet solution, which is described in Chapter 2 [4]. This makes it substantially more appropriate for real-world use in dietary assessment than the previously-developed application. Apart from Vid, research into mobile applications for food image recognition has also been performed by other research groups [8], [33]. The performance of Vid cannot easily be compared to other mobile solutions, as they were trained and tested on different food image datasets.

In the future, the main goal will be to upgrade the Vid application and the solution, presented in Chapter 4, by training the DCNN model on additional real-world food and drink images. This would allow it to recognize more food and drink items, as well as achieve a higher precision. Additionally, food volume estimation can be implemented in the application to automate that aspect of food tracking as well. Finally, with the goal of achieving an even higher precision, the predictions from the DCNN model can be tailored to each individual user by favoring food items that the user photographed in the past, or are similar to those items. This approach would need to be researched to conclude whether it would indeed lead to more accurate results.

The research work, performed in the scope of this doctoral dissertation, resulted in

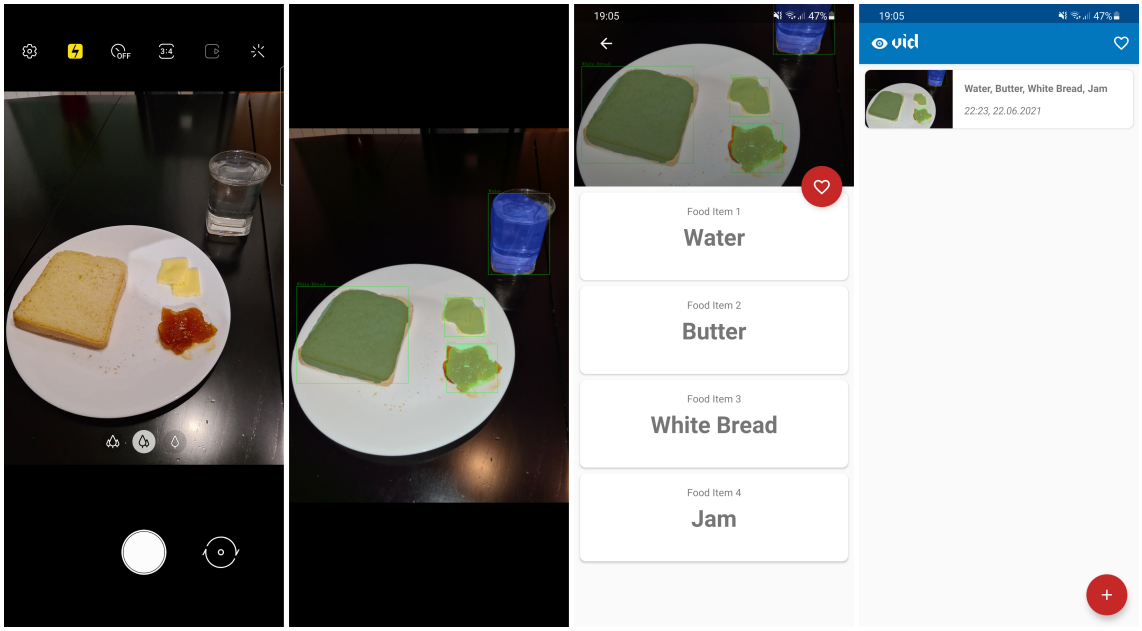

Figure 5.1: Screenshots of the Vid mobile application. The first screenshot from the left displays the photographing process, the second screenshot displays the segmented image, the third screenshot displays a thumbnail of the segmented image, along with the names of the recognized food and drink items, whereas the fourth screenshot displays the user's food diary.

the publications, included in Chapters 2, 3, and 4, along with part of the publication, mentioned in Chapter 1 [20]. These publications were met with a positive response by the scientific community, as they were cited a total of 138 times at the time of writing, according to Google Scholar [16]. Several researchers from globally established research institutions cited these works, such as Imperial College London [34], Tsinghua University [35], Cornell University [36], Columbia University [37], University of California, San Diego [38], KU Leuven [39], and many others. Additionally, researchers from some of the largest technology companies in the world cited the above publications, including Google [36], [40], Amazon [41], Facebook [42], and Samsung [43], [44].

The largest portion of citations was achieved by the publication, included in Chapter 2 [4], with 111, according to Google Scholar [16]. This is partially due to the fact that it was published first, but also possibly due to the fact that it introduced a novel DCNN architecture for food image recognition, which is characterized by a fast training time relative to its classification accuracy on food images. This, along with the fact that it was the first solution to recognize images of drinks, made it a significant part of the research into more effective food image recognition approaches, which is evidenced by the number of citations.

The second-highest cited work is the publication that was published second [17], and is included in Chapter 3. It achieved 24 citations, according to Google Scholar [16]. The other two publications [18], [20] were published very recently—both in 2021—so they only have a total of 3 citations, according to Google Scholar [16]. A direct result of the research work is also the finalist selection for the 2019 DSM Bright Science Award [21] and the high FRC placement [19], as described above.

Chapter 6

Conclusions

The research on food and drink image detection, recognition, and segmentation using deep convolutional neural networks was performed over the span of multiple years, and it culminated in this doctoral dissertation. It can be broken down into three phases: first, there was the development of single-output food image recognition solutions [4], followed by the development of fake-food image segmentation approaches [17], and finally, there was the development of real-food image segmentation solutions, with validation on real-world images of food and drink items [18]. Apart from the research findings and publications, one of the final results is a mobile application called Vid, which is able to automatically recognize food and drink items by simply taking a photograph with a smartphone. This application uses the best developed solution, which is the HTC ResNet food and drink image segmentation and classification approach [18].

Individual research steps and solutions are described in the chapters, preceding this one. Overall, the main conclusions from researching the defined hypotheses are that food image recognition can be performed in multiple ways, and that new approaches can be developed that improve upon previous solutions. Food image recognition with a single output is suitable only for more straightforward approaches as it is limited to one food or drink item per image, thus limiting its real-world usability. Food image segmentation is significantly more promising, although it is very challenging to acquire a food image dataset suitable for this purpose, as most food images are not annotated on a pixel level, which is a requirement for fine-grained segmentation.

In this regard, the Food Recognition Challenge [19] was an important milestone, as, to the best of the author's knowledge, it offered the first publicly available large-scale dataset of real-world food images, annotated on a pixel level. This allows the evaluation and comparison of recognition solutions on real-world food images, which is needed to accelerate progress towards more accurate applications for dietary assessment. These applications are able to alleviate issues with manual methods and speed up dietary assessment, making them valuable in the pursuit of improving public health through improved diets.

The research work, presented in this dissertation, serves to prove the viability of using DCNNs for food image recognition. It contributed multiple approaches to this research field [4], [17], [18], and the dissertation conclusions are in line with other research [45]. By achieving the second-best result in the second round of the aforementioned FRC, it was also proven to be competitive with other state-of-the-art approaches in the field [19].

There are two major paths forward for the food image recognition field. First, as many different food image datasets as possible need to be gathered, cleaned, and annotated. The size and diversity of the datasets that are used to train DCNN models are crucial to achieve a high classification accuracy. Therefore, if food image datasets are acquired, processed, merged, and standardized on a global scale, it has the potential to allow for the training

of significantly more accurate DCNN models that might even exceed human recognition in the future, which is not yet the case [46]. Work has been done in this direction already [47], but considerably more needs to be done to reach optimal solutions.

Second, because food image segmentation solutions can classify each pixel of the image, the result, apart from the names of food and drink items, is also the number of pixels that corresponds to each item. Consequently, it is possible to estimate food volume using this approach. Due to the changing photographing distance, doing so directly is a challenge, and it often relies on estimations based on objects of standard size that are present in the image. These can include cutlery, plates, glasses, etc. Another approach to food volume estimation is to use a reference object, although that requires significantly more effort on the part of the individual. Food volume estimation solutions are already being researched [48], but further work needs to be done to achieve a satisfactory accuracy on the large number of food items that will presumably be found in future food image datasets.

In conclusion, food image recognition is a very promising field that has the potential to substantially transform how dietary assessment is performed in practice. Perhaps even more significantly, because it lowers the barrier to entry, it can potentially enable the dietary assessment of a large portion of the general population, thus contributing to improvements in public health.

References

[1] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt: IEEE, 2009, pp. 285–288.

[2] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A food recognition system for diabetic patients based on an optimized bag-of-features model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1261–1271, 2014.

[3] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA: IEEE, 2010, pp. 2249–2256.

[4] S. Mezgec and B. K. Seljak, "NutriNet: A deep learning food and drink image recognition system for dietary assessment," *Nutrients*, vol. 9, no. 7, p. 657, 2017.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[7] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA: IEEE, 2018, pp. 567–576.

[8] H. Zhao, K.-H. Yap, A. C. Kot, and L. Duan, "JDNet: A joint-learning distilled network for mobile visual food recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 665–675, 2020.

[9] W. Shimoda and K. Yanai, "Weakly-supervised plate and food region segmentation," in *Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME)*, London, UK: IEEE, 2020, pp. 1–6.

[10] A. Singla, L. Yuan, and T. Ebrahimi, "Food/non-food image classification and food categorization using pre-trained GoogLeNet model," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa)*, Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 3–11.

[11] S. Christodoulidis, M. Anthimopoulos, and S. Mougiakakou, "Food recognition for dietary assessment using deep convolutional neural networks," in *New Trends in Image Analysis and Processing – 18th International Conference on Image Analysis and Processing (ICIAP) 2015 Workshops*, Genoa, Italy: Springer International Publishing, 2015, pp. 458–465.

[12] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: A new dataset, experiments, and results," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 588–598, 2017.

[13] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt: IEEE, 2009, pp. 289–292.

[14] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo (ICME)*, Melbourne, VIC, Australia: IEEE, 2012, pp. 25–30.

[15] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, "ISIA Food-500: A dataset for large-scale food recognition via stacked global-local attention network," in *Proceedings of the 28th ACM International Conference on Multimedia (MM)*, Seattle, WA, USA: Association for Computing Machinery, 2020, pp. 393–401.

[16] (2021). "Google Scholar: Simon Mezgec," [Online]. Available: `https://scholar.google.com/citations?user=s3fRW2gAAAAJ` (visited on 10/25/2021).

[17] S. Mezgec, T. Eftimov, T. Bucher, and B. K. Seljak, "Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment," *Public Health Nutrition*, vol. 22, no. 7, pp. 1193–1202, 2019.

[18] S. Mezgec and B. K. Seljak, "Deep neural networks for image-based dietary assessment," *Journal of Visualized Experiments*, vol. 169, e61906, 2021.

[19] (2021). "AIcrowd Food Recognition Challenge," [Online]. Available: `https://www.aicrowd.com/challenges/food-recognition-challenge` (visited on 05/14/2021).

[20] N. V. Matusheski, A. Caffrey, L. Christensen, S. Mezgec, S. Surendran, M. F. Hjorth, H. McNulty, K. Pentieva, H. M. Roager, B. K. Seljak, K. S. Vimaleswaran, M. Remmers, and S. Péter, "Diets, nutrients, genes and the microbiome: Recent advances in personalised nutrition," *British Journal of Nutrition*, pp. 1–9, 2021.

[21] (2019). "2019 DSM Bright Science Awards," [Online]. Available: `http://www.fens2019.org/dsm-science-technology-award-2019` (visited on 05/16/2021).

[22] (2019). "13th European Nutrition Conference," [Online]. Available: `http://www.fens2019.org` (visited on 06/27/2021).

[23] S. Mezgec and B. K. Seljak, "Using deep learning for food and beverage image recognition," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, 2019, pp. 5149–5151.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA: Association for Computing Machinery, 2012, pp. 1097–1105.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, 2015, pp. 1–9.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 770–778.

[27] T. Bucher, K. van der Horst, and M. Siegrist, "Fruit for dessert. How people compose healthier meals," *Appetite*, vol. 60, pp. 74–80, 2013.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, 2015, pp. 3431–3440.

[29] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid task cascade for instance segmentation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, 2019, pp. 4974–4983.

[30] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment," in *Proceedings of the 14th International Conference on Smart Homes and Health Telematics (ICOST)*, Wuhan, China: Springer International Publishing, 2016, pp. 37–48.

[31] H. Hassannejad, G. Matrella, P. Ciampolini, I. D. Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very deep convolutional networks," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa)*, Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 41–49.

[32] (2021). "Android," [Online]. Available: `https://www.android.com` (visited on 06/24/2021).

[33] R. Tanno, K. Okamoto, and K. Yanai, "DeepFoodCam: A DCNN-based real-time mobile food recognition system," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa)*, Amsterdam, The Netherlands: Association for Computing Machinery, 2016, p. 89.

[34] B. Cobley and D. Boyle, "OnionBot: A system for collaborative computational cooking," *arXiv*, arXiv:2011.05039, 2020.

[35] K. Bi, T. Qiu, and Y. Huang, "A deep learning method for yogurt preferences prediction using sensory attributes," *Processes*, vol. 8, no. 5, p. 518, 2020.

[36] P. Kaur, K. Sikka, W. Wang, S. Belongie, and A. Divakaran, "FoodX-251: A dataset for fine-grained food classification," *arXiv*, arXiv:1907.06167, 2019.

[37] P. Y. Chen, J. D. Blutinger, Y. Meijers, C. Zheng, E. Grinspun, and H. Lipson, "Visual modeling of laser-induced dough browning," *Journal of Food Engineering*, vol. 243, pp. 9–21, 2019.

[38] D. Spruijt-Metz, C. K. F. Wen, B. M. Bell, S. Intille, J. S. Huang, and T. Baranowski, "Advances and controversies in diet and physical activity measurement in youth," *American Journal of Preventive Medicine*, vol. 55, no. 4, e81–e91, 2018.

[39] S. Van Asbroeck and C. Matthys, "Use of different food image recognition platforms in dietary assessment: Comparison study," *JMIR Formative Research*, vol. 4, no. 12, e15602, 2020.

[40] Q. Thames, A. Karpur, W. Norris, F. Xia, L. Panait, T. Weyand, and J. Sim, "Nutrition5k: Towards automatic nutritional understanding of generic food," *arXiv*, arXiv:2103.03375, 2021.

[41] A. Salvador, E. Gundogdu, L. Bazzani, and M. Donoser, "Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning," *arXiv*, arXiv:2103.13061, 2021.

[42] A. Salvador, M. Drozdzal, X. Giró-i-Nieto, and A. Romero, "Inverse cooking: Recipe generation from food images," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, 2019, pp. 10 453–10 462.

[43] J. Li, R. Guerrero, and V. Pavlovic, "Deep cooking: Predicting relative food ingredient amounts from images," in *Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management (MADiMa)*, Nice, France: Association for Computing Machinery, 2019, pp. 2–6.

[44] J. Li, F. Han, R. Guerrero, and V. Pavlovic, "Picture-to-amount (PITA): Predicting relative ingredient amounts from food images," in *Proceedings of the 2020 International Conference on Pattern Recognition (ICPR)*, Milan, Italy: IEEE, 2021, pp. 10 343–10 350.

[45] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, "Application of deep learning in food: A review," *Comprehensive Reviews in Food Science and Food Safety*, vol. 18, no. 6, pp. 1793–1811, 2019.

[46] P. Furtado, M. Caldeira, and P. Martins, "Human visual system vs convolution neural networks in food recognition task: An empirical comparison," *Computer Vision and Image Understanding*, vol. 191, p. 102 878, 2020.

[47] G. Ciocca, P. Napoletano, and R. Schettini, "CNN-based features for retrieval and classification of food images," *Computer Vision and Image Understanding*, vol. 176–177, pp. 70–77, 2018.

[48] F. P.-W. Lo, Y. Sun, J. Qiu, and B. Lo, "Food volume estimation based on deep learning view synthesis from a single depth map," *Nutrients*, vol. 10, no. 12, p. 2005, 2018.

Bibliography

Publications Related to the Dissertation

Journal Articles

S. Mezgec and B. K. Seljak, "NutriNet: A deep learning food and drink image recognition system for dietary assessment," *Nutrients*, vol. 9, no. 7, p. 657, 2017.

S. Mezgec, T. Eftimov, T. Bucher, and B. K. Seljak, "Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment," *Public Health Nutrition*, vol. 22, no. 7, pp. 1193–1202, 2019.

S. Mezgec and B. K. Seljak, "Deep neural networks for image-based dietary assessment," *Journal of Visualized Experiments*, vol. 169, e61906, 2021.

N. V. Matusheski, A. Caffrey, L. Christensen, S. Mezgec, S. Surendran, M. F. Hjorth, H. McNulty, K. Pentieva, H. M. Roager, B. K. Seljak, K. S. Vimaleswaran, M. Remmers, and S. Péter, "Diets, nutrients, genes and the microbiome: Recent advances in personalised nutrition," *British Journal of Nutrition*, pp. 1–9, 2021.

Conference Papers

S. Mezgec and B. K. Seljak, "Using deep learning for food and beverage image recognition," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, 2019, pp. 5149–5151.

Other Publications

Conference Papers

S. Mezgec and P. Rogelj, "Traffic sign symbol recognition with the D2 shape function," in *Proceedings of the Middle-European Conference on Applied Theoretical Computer Science (MATCOS) 2013*, Koper, Slovenia: University of Primorska Press, 2016, pp. 35–38.

Biography

Simon Mezgec earned his Bachelor's and Master's Degrees in Computer Science from the University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia, in 2011 and 2014, respectively. Additionally, he completed the CS188.1x: Artificial Intelligence online course by the University of California, Berkeley, USA, with distinction, and he holds the Android Developer Nanodegree by Udacity. With this dissertation, he is pursuing a Ph.D. Degree in Information and Communication Technologies at the Jožef Stefan International Postgraduate School, Ljubljana, Slovenia.

During the three stages of his studies, he achieved grade averages of 9.71, 9.83, and 9.92, respectively. He was a recipient of the Faculty of Mathematics, Natural Sciences and Information Technologies Scholarship, the Zois National Scholarship for Gifted Students, the Jožef Stefan Institute Scholarship, and the World Federation of Scientists National Scholarship. He was also one of the four finalists for the 2019 DSM Bright Science Award and a two-time recipient of the Faculty of Mathematics, Natural Sciences and Information Technologies Outstanding Achievement Award. In the second round of the AIcrowd Food Recognition Challenge, he achieved second place.

He is currently working as an associate at the Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia. There, he has been working on several projects related to automatic food recognition. His research interests include computer vision, image processing, and deep learning. More specifically, he is working on food and drink image detection, recognition, and segmentation, for which he is using deep convolutional neural networks.