STAG: Smart Tools and Applications in Graphics (2021)
P. Frosini, D. Giorgi, S. Melzi, and E. Rodolà (Editors)

SlowDeepFood: a food computing framework for regional gastronomy.

N. U. Gilal[1] and K. Al-Thelaya[1] and J. Schneider[1] and J. She[1] and M. Agus[1]

[1] College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

Abstract

Food computing recently emerged as a stand-alone research field, in which artificial intelligence, deep learning, and data science methodologies are applied to the various stages of food production pipelines. Food computing may help end-users in maintaining healthy and nutritious diets by alerting of high caloric dishes and/or dishes containing allergens. A backbone for such applications, and a major challenge, is the automated recognition of food by means of computer vision. It is therefore no surprise that researchers have compiled various food data sets and paired them with well-performing deep learning architecture to perform said automatic classification. However, local cuisines are tied to specific geographic origins and are woefully underrepresented in most existing data sets. This leads to a clear gap when it comes to food computing on regional and traditional dishes. While one might argue that standardized data sets of world cuisine cover the majority of applications, such a stance would neglect systematic biases in data collection. It would also be at odds with recent initiatives such as SlowFood, seeking to support local food traditions and to preserve local contributions to the global variation of food items. To help preserve such local influences, we thus present a full end-to-end food computing network that is able to: (i) create custom image data sets semi-automatically that represent traditional dishes; (ii) train custom classification models based on the EfficientNet family using transfer learning; (iii) deploy the resulting models in mobile applications for real-time inference of food images acquired through smart phone cameras. We not only assess the performance of the proposed deep learning architecture on standard food data sets (e.g., our model achieves 91.91% *accuracy on ETH's Food-101), but also demonstrate the performance of our models on our own, custom data sets comprising local cuisine, such as the Pizza-Styles data set and GCC-30. The former comprises 14 categories of pizza styles, whereas the latter contains 30 Middle Eastern dishes from the Gulf Cooperation Council members.*

CCS Concepts
• ***Human-centered computing*** → *Ubiquitous and mobile computing;* • ***Computing methodologies*** → ***Scene understanding;*** *Object recognition;*

1. Introduction

Driven by the progress in artificial intelligence and deep learning, a recent, enormous boost in computer vision has triggered the emergence of a plethora of applications related to image analysis that are fundamentally changing the way people interact with multimedia systems. Among those, a task that is becoming especially common is the automatic recognition and classification of images taken casually. Applications based on the resulting 'smart cameras' are nowadays ubiquitous in the mobile ecosystem. As of writing, most vendors have started equipping their high-end smartphones with hardware-bases inference accelerators and have created artificial intelligence frameworks that optimize picture quality by analyzing the content of the scene to be captured and adjusting camera settings accordingly. The very same computational infrastructure also fuels a rapid evolution in the application domain, driven by the automatic recognition of objects, scenes, persons, animals, etc.

In this application domain, food computing [MJL*19] is a field that recently emerged and gained high popularity quickly. A main challenge to be addressed in this context is the automated understanding of food images, especially with respect to detection, segmentation, and analysis of food on trays, measuring portions, estimation of quality, nutritional value, and/or caloric content. The prominent goals of most efforts addressing this challenge are health-related targets, such as providing nutritional recommendations to users (driven by nutrition and Calorie estimation) as well as screening food for ingredients or potential allergens to meet the users' dietary requirements.

While labeled data sets containing samples of international cuisines are available to train deep models, the same is not true for traditional and regional specialties. As a result, training deep neural networks using supervised learning to classify food according to internationally recognized taxonomies has become a relatively simple exercise. At the same time, the lack of data renders the same task impossible for regional and traditional cuisines. However, the

DOI: 10.2312/stag.20211476

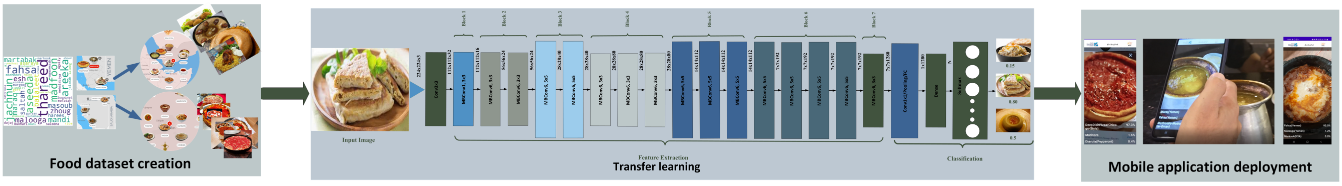


Figure 1: ***SlowDeepFood.*** *Our end-to-end food computing framework is able to create regional food data sets semi-automatically (left). It also creates custom classification models based on transfer learning (center) that can be deployed to mobile devices and integrated into real-time applications (right).*

latter cuisines are gaining more and more interest in a trend to move away from fast food and back to more traditional ways of food preparation. Various international organizations like IGCAT (International Institute of Gastronomy, Culture, Arts and Tourism)[1] and SlowFood[2] regularly promote initiatives to raise awareness of the importance of cultural and food uniqueness in an effort to preserve distinctive food cultures. Inspired by such efforts, we address the need to fill gastronomy representation gaps in public domain data. We believe that therefore there is a need for customized data relevant to different gastronomic cultures, such as outlined in this paper. We also would argue that the availability of such data will contribute to the aims of the aforementioned organizations, especially concerning: (i) preventing the disappearance of local food traditions, (ii) stimulating creativity, (iii) educating for better nutrition, and (iv) improving sustainable tourism standards [Sim12]. With respect to the latter point, we would also argue that being able to identify commonly encountered dishes of a globalized cuisine provides less value than being able to identify and describe "exotic" dishes, e.g., such as encountered on travels, with which one is unfamiliar.

In order to provide technological support to initiatives promoting worldwide food diversity, we present *SlowDeepFood*, an end-to-end deep learning framework for food computing that is able to efficiently target custom local gastronomies. The proposed framework integrates the following technical contributions:

- a semi-automatic processing pipeline for the fast creation of specific food data sets representing gastronomic regions or specific food categories (Sec. 3.2). Using Selenium's Python bindings[3] (a portable framework for testing web applications) combined with the chromedriver library, the pipeline is able to efficiently create medium complexity, customized data sets for automatic food classification in a reasonable time frame (around 30 minutes per food category) ;
- an effective deep learning framework utilizing the concept of transfer learning and built on top of modern EfficientNet architectures. Our framework is able to create lightweight models to be used in mobile applications (Sec. 3.3). The proposed transfer learning strategy is able to train accurate classification models in reasonable time (max. 6 hours for EfficientNet-B4 on Food-101) on recent commodity GPUs;
- a prototype mobile application for real-time food classification that integrates the models trained on our specific custom data sets (Sec. 3.4). The application uses the smartphone camera for performing real-time picture acquisition and predicts the top k categories at a rate of several inferences per second using our GPU-trained classifier.

We validate the deep learning framework on the most popular general food data set (namely ETH-Food101 [BGVG14]), for which we obtain performance comparable to state-of-the-art methods [JMLL20] (that is, 91.91% top-1 accuracy for EfficientNet-B4). We finally demonstrate the proposed technology on two newly created custom data sets, namely Pizza-Styles containing 14 styles of pizza, and GCC-30, containing 30 traditional dishes from Arabic Gulf Countries. To this end, we report on accuracy of our classification models and on preliminary evaluation of mobile applications in typical usage scenarios in traditional restaurants.

2. Related work

Our work deals with food computing and deep learning applied to image classification. Since a full review of the rapidly developing field of food computing is beyond the scope of this work, we would like to refer readers to surveys [MJL*19,ZZL*19]. In the following we overview the main problems, the applications currently available, the deep learning frameworks currently used for image classification, as well as the data sets available for deep learning.

Food computing. Given the increasing availability of public data and the evolution of AI technologies, a new computing field named automated food analysis has recently emerged. The main challenges addressed by the field are related to the classification and recognition of food images, and various methods have been proposed along the last decade. Advances in machine learning technologies have extensively improved the accuracy of object detection, identification, and recognition from single pictures. The field has particularly benefited of deep learning and convolutional neural networks (CNNs). The latter have been applied, among others: food recognition [SSC*20, ZYK21], food segmentation [GdMWP*20, MSFV*21], food-tray analysis [Pop20], food classification [CMN20, SGH*21], ingredient recognition [CZN*20], food quality estimation [LNC20], calorie counting [LAM*20, SDBJ21] and portion estimation [JQL*20].

[1] https://www.europeanregionofgastronomy.org/, accessed 11.Sep.2021
[2] https://www.slowfood.com/, accessed 11.Sep.2021
[3] https://selenium-python.readthedocs.io/, accessed 16.Sep.2021

Improvements in the performance of such tasks have in turn enabled various applications to fuse multimedia towards a variety of health-related targets: to provide nutritional guidelines to users, e.g., calories and nutrition estimation [LSV*20], food recommendation related to specific health conditions [RMK20] and automatic dietary monitoring of canteen customers [CNS15a].

In this paper, we propose a framework for supporting Slow Food initiatives aimed at the protection of local and traditional recipes. Our framework fuses a semi-automatic preprocessing pipeline for creating custom data sets related to specific gastronomy regions or food categories with a deep learning architecture based on the recent EfficientNet family of convolutional neural networks [TL19].

Convolutional neural networks in mobile systems. Convolutional neural networks (CNNs) are, at their core, deep artificial neural networks that try to emulate the behavior of the visual cortex in the human nervous system. They have become extremely popular in the visual computing community over the last decade for being able to solving a wide range of problems related to image processing [LHL16]. The success of CNNs can be largely attributed to two trends. Firstly, fueled by increasing computational capabilities as well as architectural advances in deep learning, models became increasingly bigger. The reason was an attempt to achieve better accuracy on ImageNet competitions, starting from AleXNet [KSH12] that won the 2012 ImageNet competition, up to Squeeze-and-Excitation Networks [HSS18] and GPipe [HCB*19], winning the same competition in recent years. Secondly, with the rise of smart phones to ubiquity, a demand for the design of CNNs that perform efficient inference on such traditionally limiting platforms arose. This demand has been addressed by hand-crafted models such as SqueezeNets [SdSZ*18] and MobileNets [QZC*18], or even fully automated neural architecture search (NAS). NAS seeks to optimize the hyperparameters of CNNs (depth, size, activation functions, etc.) using modular building blocks as the underlying architecture. Notably, this approach has led to the Efficientnet family of CNNs [TL19]. In the context of mobile-friendly CNNs, the focus of further improvements is usually on the inference speed. Some recent architectures like EfficientNet-X [LTP*21], for instance, try to improve GPU and/or TPU inference speed. Other improvements focus on the training speed, such as BoTNets [SLP*21]. Pertaining to food recognition, there have been various recent attempts to utilize CNNs for training multi-label classifiers for predicting the kind of food. Examples include GoogLeNet [MJR*15] and residual networks [MFM18]. Very recently, Min et al. [MLW*20] proposed a stacked global-local attention network, which consists of two subnetworks for food recognition, while Jiang et al. [JMLL20] designed a Multi-Scale Multi-View Feature Aggregation (MSMVFA) scheme. The idea is to aggregate high-level semantic features, mid-level attribute features and deep visual features into a unified representation. The problem of the latter architecture is that it needs additional ingredient knowledge to obtain mid-level attribute representation via ingredient-supervised CNNs. In general, all discussed methods are not optimized to be used in a mobile setting. In contrast, our work explicitly focuses on proposing a light-weight architecture derived from EfficienNet family of CNNs [TL19] to enable mobile inference.

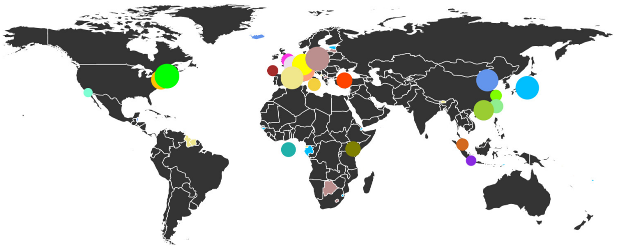


Figure 2: ***Geographic distribution of food data sets.*** *Many important gastronomy areas are not adequately represented. Also refer to our list of publicly available, geo-referenced food data sets at https://slowdeepfood.github.io/datasets/.*

Food data sets. Concurrently to the evolution of architectural advances of deep models, not only in the field of food computing, the last years have also witnessed an accelerated growth of a food image ontology coupled with sample data sets to support large-scale food recognition. These data sets now form the benchmarks and test-beds for algorithms and models in food computing. Among them, the most popular ones are: ETH's Food101 [BGVG14], containing 101,000 images representing 101 food categories, UNIMIB2016 [CNS15b] containing 1,027 canteen trays for a total of 3,616 food instances belonging to 73 food classes (to be used for object detection and dietary assessment), UNICT-FD889 [FAS14], containing over 800 distinct plates of food, ISIA Food-500 [MLW*20] with, as the name suggests, 500 categories and 399,726 images, and UEC-FoodPix [OY21], a large-scale food image segmentation data set comprising 10,000 food images with segmentation masks. In this work, we carry out a critical analysis of recently published data sets according to the represented cultural and regional environments to create a Geo-referenced classification. We published a web resource listing of the data sets currently available [4]. As shown in Fig. 2, the geographic distribution of data sets available is far from uniform, critically under-representing many important areas: most data sets were created for stressing automatic processing methods and they are too general for being applied to different culinary styles, methods and regions. To overcome this limitation, we propose a semi-automatic processing pipeline for creating databases representing strictly localized regions of gastronomy. We demonstrate the potential of our approach by targeting some areas that have not been considered for benchmark data as of writing.

3. SlowDeepFood Framework

3.1. Overview

We postulate the following requirements for our framework:

- **R1. Fill the striking representation gaps of traditional cuisines in currently available data sets.** Doing so will boost the impact of initiatives seeking to protect local food traditions and remedy the current limitations of available data sets.

[4] https://slowdeepfood.github.io/datasets/, accessed 17.Sep.2021.

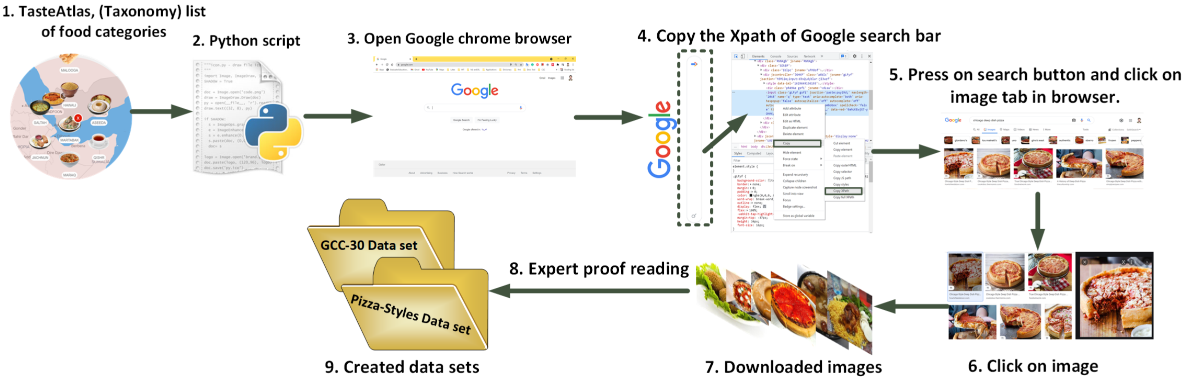

1. TasteAtlas, (Taxonomy) list of food categories
2. Python script
3. Open Google chrome browser
4. Copy the Xpath of Google search bar
5. Press on search button and click on image tab in browser.
6. Click on image
7. Downloaded images
8. Expert proof reading
9. Created data sets
GCC-30 Data set
Pizza-Styles Data set

Figure 3: ***Data set creation pipeline.*** *Starting from a taxonomy of food specialties, a Python script takes control of Google's images search engine. Through automated scroll and click operations, we are to download hundreds of images in a few minutes. This is followed by manual proofreading to clean the data set.*

- **R2. Perform fine-grained classification dishes.** Food recognition is a difficult task. Many dishes share common features, and intra-class variance may often be larger than inter-class variance. Therefore, standard models need to be customized, especially for custom data sets representing local traditional food specialities for which small geographic displacements may have great impact on the visuals of the final dish.
- **R3. Enable isolated mobile inference.** Deploying most architectures on mobile devices *without* relying on a cloud back-end ("isolated") suffer from being overly optimized for accuracy performance and not for inference speed. The idea of performing isolated inference is gaining more and more attention, though, and benefits tourists requiring apps that are efficient in terms of the use of memory, bandwidth, battery power, and time (latency) during inference.

The SlowDeepFood framework we propose satisfies the aforementioned requirements by designing, implementing, and integrating the following components (for a schematic depiction see Fig. 1):

- **Semi-automatic custom data set creation:** starting from a list of predefined food categories, obtained by gathering input from domain experts, books, and web resources, we use automatic scripts to download hundreds of images for each categories on popular search engines. After that, again with the help of experts, we perform manual proof-reading to clean the image collections.
- **Classification through transfer learning on CNNs:** in order to solve fine-grained classification problems, we fine tune a family of pre-trained big CNNs through a transfer learning process. Transfer learning starts with a network trained on a different task (e.g., ImageNet), which is used as a feature extractor. A classification "adaptor", called head, (c.f. Fig. 4) is trained first and for only a few epochs to convert the commonly more than 1,000 features to the number of classes at hand. Then, the full network including the "body" is fine tuned for another few epochs. The underlying assumption is that the features one seeks to train the network to are very similar between tasks, and much of a more expensive, previous training session can be recycled. In this work, we consider the popular family of EfficienNet networks [TL19].
- **Isolated mobile deployment:** for isolated deployment on mobile CPUs, GPUs, or Edge TPUs we convert our models to make best use of the computational resources available on the mobile device (i.e., we use Efficientnet light, targeting the TensorFlow Lite framework). This optimizes the use of CPU, memory, and battery and it minimizes inference time. On top of that, we developed a prototype application for real-time classification of images gathered from the mobile's back camera.

In the following subsections we detail the components of the SlowDeepFood framework.

3.2. Data set creation

A pre-requisite for any classification model based on computer vision technologies is an image data set for training and testing. Considering the task of assembling a data set on the specialties of a regional cuisine, we identify the following steps:

- individuate a taxonomy composed of a list of categories that we want to discriminate;
- find a sufficient number of representative pictures for each category to form a training and testing set;
- proofread the database with help of experts to filter out wrongly labeled pictures.

For the first step, many resources may be used, ranging from online resources related to food categories to indication from food experts, books, etc. In this work, we first considered the indications from food experts, integrated with the information extracted from a general public resource called TasteAtlas [5] which features an interactive global food map with dish icons shown in their respective regions. TasteAtlas purportedly contains nearly ten thousand dishes (see Fig. 3).

After compiling the taxonomy containing the food category list,

[5] https://www.tasteatlas.com/, accessed 11.Sep.2021

we need to find a representative set of pictures online for each item in a given set of regional dishes or specialties. To do so, we developed a semi-automatic Python tool using Python's Selenium bindings. Selenium is a framework to conduct unit tests on scripts embedded in webpages. We use the chromedriver library to run the scripts on webpages and Selenium to feed automated events into the browser. Our tool opens a tab on the browser and then performs an automatic search on Google's image databases. After that, we feed automated scroll and click operations to the browser, allowing for the rapid download of hundreds of images (see also Fig 3). Since automatically downloaded data has a significant noise level (i.e., pictures are wrongly labelled), we perform expert proof-reading of the entire collection. For each food category we thus manually check all pictures and exclude incorrectly labelled ones. In Sec. 4 we report statistics gathered during the creation of a custom data sets for Pizza specialties ("Pizza-Styles") and for traditional dishes of Middle Eastern Gulf countries ("GCC-30").

3.3. Classification model

Similar to other computer vision applications, we exploit the transfer learning paradigm (see Fig. 4) for our food image classification problem. Transfer learning starts with an existing model that was pre-trained on a popular generic data set (in our case ImageNet). The bulk of this model is used as a feature extractor ("body"). We then add dense layer with softmax activation function ("head") to convert the generic features into the probabilities for each of the categories. The network is then fine tuned in two steps: First, the head is trained while body weights are kept static. This is to avoid back-propagating the untrained weights of the head into the body. In a second step, head and body are fine tuned to improve performance. We use the EfficientNet family [TL19] as the body of our architecture for two main reasons:

- **accuracy:** as of writing, EfficientNet models achieve the highest accuracy on the ImageNet data set. They have replaced ResNet-50 for many deep learning tasks[6];
- **scalability:** EfficientNets were generated according to an efficient scaling method, called compound scaling. This scaling describes optimal scaling of network depth, width, and input image resolution to maximize accuracy while minimizing inference cost.

EfficientNet's compound scaling is parameterized by a single scaling factor ϕ, deriving the scaling for depth, width, and input image resolution as follows. The network depth is scaled by α^{ϕ}, the width by β^{ϕ}, and the resolution by γ^{ϕ}. The values of α, β, γ are fixed, and obtained through grid search optimization in a way that for any new ϕ, the total number of floating point operations (FLOPS) for inference will increase by 2^{ϕ}. The baseline network is called EfficientNet-B0 (depicted in Fig. 4), and it is characterized by:

- **MBConv building blocks** [SHZ*18], residual blocks using an narrow-wide-narrow structure. In the case of EfficientNet-B0 they are coupled with with 3×3 and 5×5 depth-wise convolutions.
- **squeeze-and-excitation (SE)** [HSS18], adding parameters to each channel of the building block that allow the network to adjust the weighting of each feature map adaptively.
- **swish activation function** [MH21], defined by $\sigma(x) := \frac{x}{1+e^{-\beta x}}$. Swish can be understood as a smooth function that interpolates (non-linearly) between a linear function (for $\beta = 0$) and the rectified linear unit (ReLU) activation function (for $\beta \to \infty$). This alleviates the issue of vanishing gradient during back-propagation.

Recently, light versions of models in the EfficientNet family have been released [Liu20]. These models are optimized for EdgeTPUs as well as mobile CPUs and GPUs. EfficientNet-Lite removes the Squeeze-and-Excitation from the original EfficientNet and substitutes the swish activation function by ReLU6 (a modification of the rectified linear unit where the activation is limited to a maximum size of 6), to allow a more efficient quantization of model weights. Despite these simplifications significantly reducing the size of the models, an ImageNet accuracy above 80% is maintained.

In our framework, we use models from EfficientNet and EfficientNet-Lite families. We train the models using transfer learning as described above on our custom food data sets. Specifically, we use EfficientNet models on desktop GPUs to evaluate the nominal performance, while deploying EfficientNet-Lite classifiers on mobile devices. In Sec. 4 we report on classification performances of our models generated from both families, both for standard food data sets like Food-101 [BGVG14] and our custom data sets.

3.4. Mobile deployment

We integrated the classifiers trained on our custom food data sets into an Android application. The application classifies pictures obtained from the mobile phone's back camera in real-time, and displays the names of the most likely dishes to the user. The architecture of the application is depicted in Fig. 5 and contains the following components:

- a **camera activity process**, for controlling the device camera and to access the frames in real-time.
- **image processing tools**, needed to adapt the pictures from the camera according to the requirements of the inference model, e.g., by performing rotations, crops, centering, normalizing and resizing to the expected size (in our case 224×224) for classification. The output of this component is an image tensor that can be processed by the classifier.
- a **classifier** based on TensorFlow Lite. The classifier is obtained by quantizing the original model that was trained on a desktop GPU. This component controls our inference and can be customized by changing settings such as the number of top scored results or score thresholds.
- a **main activity process**, responsible for instantiating a properly configured classifier, to start a background thread for performing continuous inference, to overlay the classification results on top of the camera image, and a user interface. We display the top K probability values (our prototype currently uses $K = 3$) and associated dish categories.

[6] https://sotabench.com/benchmarks/image-classification-on-imagenet, accessed 11.Sep.2021

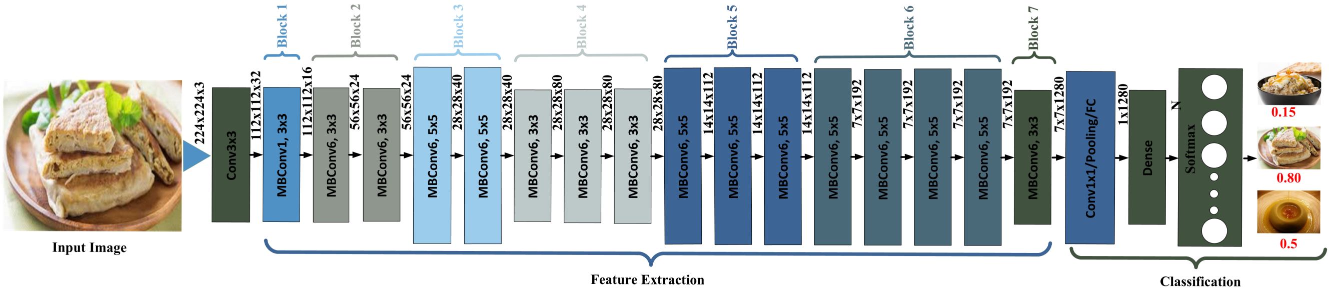
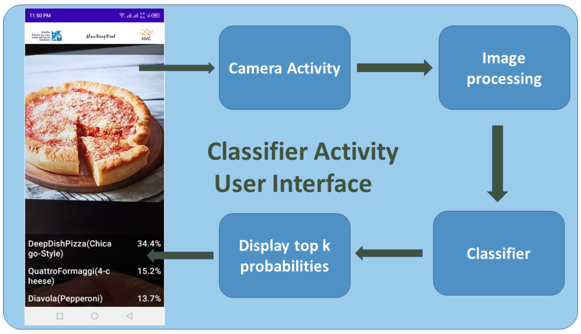


Figure 4: ***SlowDeepFood architecture.*** *Our classification model uses transfer learning on top of EfficientNet convolutional neural networks. To the "body" (feature extraction) of the network, we add a "head" (classification) consisting of an average pooling and a dense layer with softmax activation to compute the probabilities for each of the N dish categories.*



DeepDishPizza(Chicago-Style) 34.4%
QuattroFormaggi(4-cheese) 15.2%
Diavola(Pepperoni) 13.7%

Camera Activity
Image processing
Classifier Activity User Interface
Classifier
Display top k probabilities

Figure 5: ***Mobile application.*** *The custom food classification models are deployed on mobile platforms and integrated in a prototype application for performing real-time inference of pictures acquired through smartphone back camera.*

The threads are organized in a producer-consumer fashion to avoid blocking classification and adding latency, thus maximizing inference and display rate. The application is, in this way, able to provide multiple inferences per seconds (see also Sec. 4 for details about performance in mobile setting).

4. Results

In the following, we report on preliminary results obtained with the SlowDeepFood framework. We assessed the classification performance on ETH's Food101 data set [BGVG14], and compared with state-of-the-art classification methods. Using the semiautomatic processing script, we create two custom food data sets:
(1) Pizza-Styles, specializing on different traditional styles of Pizza, and, (2) GCC-30, specializing on traditional dishes from Gulf Countries.
Finally, we created two custom mobile applications for real-time recognition of Pizza versions and Gulf traditional dishes and we tested them in the wild with sessions in traditional restaurants. We plan on making the mobile application prototype, our custom data sets, as well as the source code available on github at a later stage.

Implementation details We implement the various stages of our framework's pipeline using different approaches:

- the pre-processing script is implemented in Python and uses Selenium and the chromedriver library.
- the classification models are implemented in Python using Jupyter notebooks. They use the FastAI [HG20] framework for GPU training, and TensorFlow Lite as well as Model Maker for mobile deployment.
- our mobile applications are implemented in Kotlin through based on a TensorFlow Lite image classification sample [7]. They are then deployed to Android smartphones.

Custom data set creation We tested the pipeline for fast collection of food image data sets by creating two custom data sets: Pizza-Styles focuses on variations of the popular, namesake Italian dish, whereas GCC-30 focuses on a particularly interesting cuisine (namely, Middle Eastern cuisine) that has not yet been addressed by the food computing community

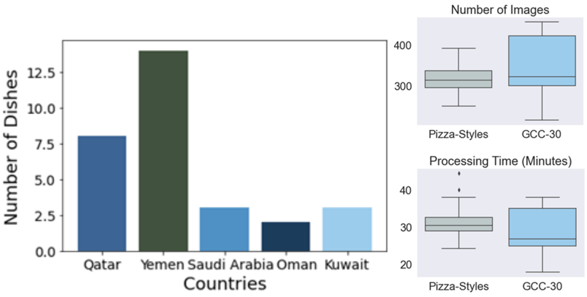


Figure 6: ***Data set creation statistics.*** *Left: histogram of specialties per country in GCC-30 data set. Top right: box-plot summarizing number of downloaded pictures per specialty for Pizza-Style and GCC-30. Bottom right: box-plot summarizing processing time in minutes per specialty for Pizza-Style and GCC-30.*

Concerning the Pizza-Styles data set, we created the taxonomy

[7] https://github.com/tensorflow/examples, accessed 11.Sep.2021

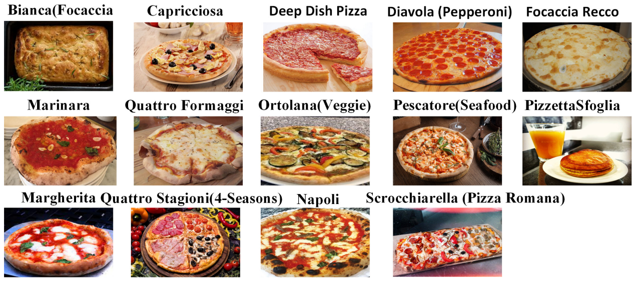


Figure 7: ***Pizza-Styles data set.*** *Our custom Pizza-Styles data set contains 14 categories representing special styles of Pizza. The training set contains 2,100 images (150 images per class), while the testing set contains 700 images (50 images per class).*

by collecting subcategories from online resources. We also included some traditional local Italian Pizza styles, like the D.O.P. Focaccia di Recco from Liguria, and Pizzetta Sfoglia from Sardinia. The number of automatic downloaded pictures was, on average, 314.75 per category. At the end of the cleaning process, we ended up with 14 Pizza subcategories, each represented by 200 images. Of these, we reserve 150 for training and 50 for testing (see Fig. 7). The overall process took an average of 31.7 minutes per category (cf. Fig. 6 for box-plots related to the number of pictures and processing times).

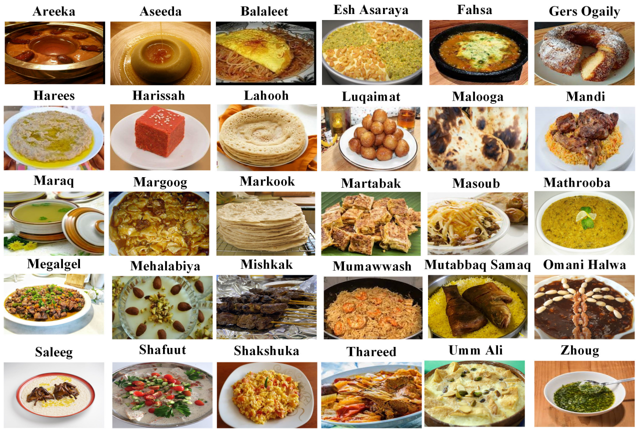


Figure 8: ***GCC-30 data set.*** *Our custom GCC-30 data set contains 30 traditional dishes from Middle East Gulf Countries. The training set contains 4,500 images (150 for each class), while the testing set contains 1,500 images (50 for each class).*

Concerning Middle East traditional cuisine, we used TasteAtlas as initial reference for collecting information about dishes, for the countries in the Arabic peninsula, namely Qatar, Yemen, Saudi Arabia, Kuwait and Oman. After compiling a first taxonomy, we gathered feedback and suggestions from local food experts in the region, as well as traditional restaurants for further refining the list of dish categories. We removed the dishes with a high pairwise similarity, as well as dishes that cannot be considered traditional and do not strictly represent the Arabic peninsula. At the end of the taxonomy creation process, we compiled a complete list of 30 dishes, representing the five Gulf countries as indicated in Fig. 6, left.

After that, we used the obtained taxonomy to search online for images representing the specialties (in most cases, we used the Arabic name of each dish to increase the chance to find pictures). In this way, we collected an average of 346.27 images per category. We then filtered and cleaned the data to arrive at a training set of 150 pictures, and a testing set of 50 pictures. The total process, including downloading and manual filtering, required an average of 28.66 minutes per category (see also Fig. 6 for detailed box-plots related to the number of pictures and the total processing time). At the end of the process, we obtained a full GCC-30 data set containing 6,000 images (cf. Fig. 8).

Classification performances To assess the performance of our transfer learning framework, we trained models based on EfficientNet family on the ETH's Food101 [BGVG14] data set, and we compared with the current state of the art. We used a desktop PC equipped with a single NVidia RTX 2080 TI 11GB DDR6 RAM. We used FastAI version 1 [HG20] to proceed with transfer learning, following the paradigm of splitting the model into body and head as previously outlined. We use FastAI's one-cycle learning policy, which schedules learning rate [Smi18] and optimizer momentum. The bounds for the learning rate schedule were estimated through a range test. This runs the model for several batches with variable rates and produces a plot depicting accuracy against learning rate, helping in selecting optimal bounds. As loss function, we used the categorical cross-entropy [ZS18] provided by FastAI. To minimize the classification loss, we used the Adam optimizer [RKK18] with default parameters ($\beta_0 = 0.9$, $\beta_1 = 0.999$, $\varepsilon = 10^{-8}$, $w_d = 0.01$). We subdivided the learning process in four stages composed by an estimation of learning rate bounds and a one-cycle training phase with a limited number of epochs (3, 7, 5 and 5, respectively, for a total of 20 epochs). After the first stage and only for the first model we performed a fast noise clean-up of the training and validation test by removing the top-losses pictures. The cleaned data was then used to train all models. We used image sizes computed according to the scaling equations for EfficientNet (ranging from 224 for EfficientNet-B0 and EfficientNet-Lite0, up to 380 for EfficientNet-B4). The batch size was derived to fit into the RAM available on our GPU (from 160 for EfficientNet-B0 and EfficientNet-Lite0, down to 24 for EfficientNet-B4). Our training times ranged from 2m30 sec for EfficientNet-Lite0 up to 18m20s for EfficientNet-B4. For evaluating the accuracy, we exploited the test-time augmentation offered by the FastAi framework.

Table 1 compares the accuracy reported by recent literature with the accuracy we achieved using EfficientNet CNNs for ETH's Food101 data set. At the time of writing, the state-of-the-art of 96.18% top-1 accuracy is obtained by a modified version of EfficientNet-V2 with a Sharpness-Aware Minimization (SAM) procedure [FKMN21]. Using our framework and despite limited computational resource, we are able to boost the original EfficientNet-B4 accuracy by 0.41% [TL19] (reducing the error by 17.4%) in less than six hours of training. Our accuracy is higher than all architectures proposed specifically for food computing so far.

It is also worth mentioning that Min et al. [MLLJ19] use ad-

Model	Top 1 Acc.	Top 5 Acc.
Deep Food [LCL*16]	77.40%	93.70%
WISeR [MFM18]	90.27%	98.71%
Inception V3 [HMC*16]	88.28%	96.88%
IG-CMAN [MLLJ19]	90.37%	98.42%
PAR-NET [QLS*19]	90.40%	-
EfficientNet-B0 [ours]	87.49%	97.09%
EfficientNet-Lite0 [ours]	85.01%	96.04%
EfficientNet-B2 [ours]	89.05%	97.70%
EfficientNet-Lite2 [ours]	86.34%	96.81%
EfficientNet-B4 orig. [TL19]	91.50%	-
EfficientNet-B4 [ours]	91.91%	98.52%
EfficientNet-L2+SAM [FKMN21]	96.18%	-

Table 1: ***Accuracy on ETH's Food101 data set.*** *Performance comparison between recent methods and the EfficientNet models used in our work. We report top-1 and top-5 accuracy, where reported by the original publication.*

ditional input metadata, namely the ingredient list associated with each image, while the WISeR architecture [MFM18] requires substantial memory amounts as well as significant computational efforts to process a single datum, thus rendering deployment on mobile devices unfeasible.

Model	Pizza-Styles Acc	GCC-30 Acc
EfficientNet-B0 [ours]	87.86%	91.67%
EfficientNet-Lite0 [ours]	85.71%	90.67%
EfficientNet-B2 [ours]	91.43%	92.67%
EfficientNet-Lite2 [ours]	87.86%	90.67%
EfficientNet-B4 [ours]	94.29%	95.33%

Table 2: ***Top-1 accuracy on Pizza-Styles and GCC-30 data sets.*** *Moving from full Efficient-B to EfficientNet Lite models shows an acceptable and graceful degradation of accuracy.*

After assessing the transfer learning framework in general, we tested various EfficientNet models on our Pizza-Styles and GCC-30 data sets. Table 2 summarizes the top-1 accuracy for both data sets. While it is clear that moving from a full desktop setup to mobile devices implies a loss of performance, we observe a very gentle degradation when migrating from the full EfficientNet-B to Lite models. We observed losses in accuracy of 2.5% (B0 to Lite0) and 3.5% (B2 to Lite2) of the top-1 accuracy, which we deem a fully acceptable trade-off for being able to perform classification anytime, anywhere. Note that the performance on GCC-30 is better as it offers more intra-class variation than Pizza-Styles. In addition we maintain a top-5 accuracy very close to 100% (omitted in the table for brevity), making our framework useful for certain dietary tasks. For instance, by combining the potential allergenes in the top-5 a conservative recommendation can be made with very high confidence. At the end of the training phase, we selected the Lite2 model for mobile deployment, since it offers a slightly better performance to inference speed trade-off on the Pizza-Styles data set.

In Fig. 9 we also provide a qualitative evaluation. There, we show various cases of successful predictions for Pizza-Style and GCC-30 using EfficientNet Lite2, the same model deployed in the smartphone app. Figure 10, left, shows the confusion matrix obtained with our EfficientNet-Lite2 model on Pizza-Styles data set: the model is able to accurately discriminate between all kind of pizza represented in the data set even if some of them have very similar appearance. In fact, Margherita and Napoli can be considered modifications of Marinara pizza, and the visual differences between them can be difficult to spot (see Fig. 10 right). Finally, Fig. 11 shows the confusion matrix for the GCC-30 data set using EfficientNet-Lite2: even if in most cases the model is able to classify specialties with high accuracy, some rice-based dishes are difficult to discriminate (as an example, see Mandi and Mutabbaq Samaq in Fig. 11).

Mobile application performances For evaluating the performance of the framework in a mobile setting, we deployed two custom classification models for our Pizza-Styles and GCC-30 data sets and created two different mobile applications. We installed them on two smartphones running the Android operating system: a Huawei P20 Pro, and a Samsung Galaxy A51. We tested the applications in the wild in two different restaurants: one traditional Yemeni restaurant and an Italian-style pizza place. This replicates one expected typical usage scenario of the technology, wherein a group of casual customers, who are not expert of a particular gastronomy, order and share a number of dishes they never saw before (see Fig. 12 left), and, naturally curious, would appreciate additional information about them. Our tests show that our mobile app enabled users to correctly recognize various traditional Yemeni specialties, ranging from Fahsa over Marqook to Mahsoub, as well as various styles of Pizza such as Diavola, Quattro Formaggi, and Ortolana (kindly also refer to the accompanying video [8]and also see Figs. 12 and13).

The video demonstrates that the EfficientNet Lite models are able to perform multiple inferences per second (we observed inference rates constantly above 20Hz). This fast inference can be further exploited to improve classification performance by aggregating multiple inference probability outcomes through dynamic filters or even more sophisticated neural network models. Such approaches would then be akin to test-time augmentation, in which the user rotates/translates/moves the camera relative to the food, thereby changing orientation and appearance under the restaurant's light sources. Voting or averaging probabilities might then lead to the same improved performance we observed for FastAI's test-time augmentation (TTA), but at a lower latency and computational cost (FastAI's TTA batches image variations into a single prediction whereas we propose to infer on single images and aggregate after the classification). We plan to investigate this interesting avenue in the future.

The technology we propose adds value to restaurant customers, since it can perform real-time classification of dishes in real-time. Users can then check ingredients and additional information online in order to assess nutritional value, potential for allergens, etc. For the future, we plan to complete the system by linking the mobile

[8] https://www.dropbox.com/s/wbabinbj5pqxavf/stag_submission5_slowdeepfood.mp4?dl=0

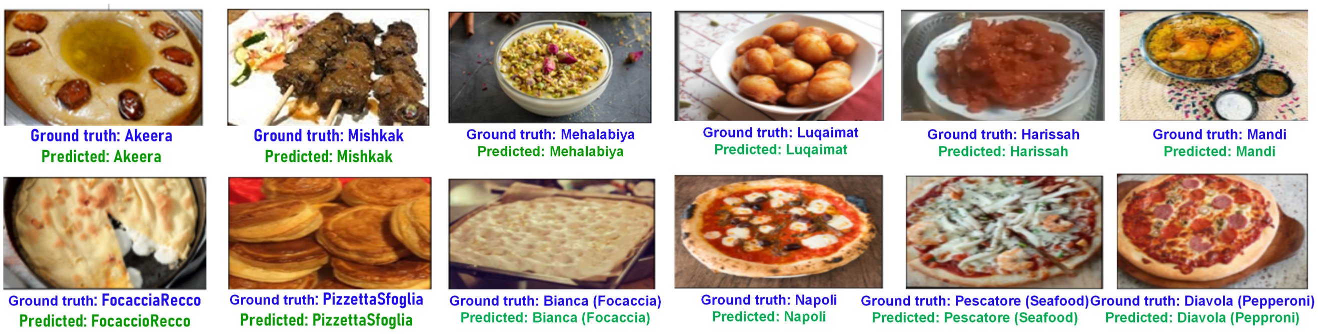


Figure 9: ***Examples of correct predictions.*** *The first row displays the predict results for GCC-30 test set, while the second row displays predictions for Pizza-Styles images.*

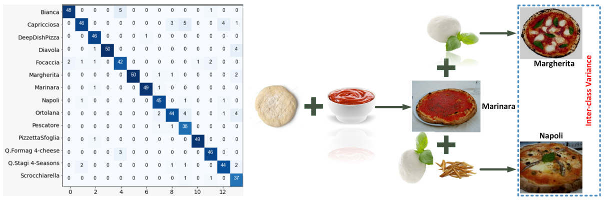


Figure 10: ***Pizza-Styles fine-grained results.*** *Left: confusion matrix shows that the model is able to accurately discriminate all kind of pizza in the data set. Right: low inter-class variation for several pizza styles stems from a similarity in preparation and ingredients. For example, starting from a Marinara style, adding mozzarella and basil results in a Margherita. Adding anchovies as yet another ingredient results in a Napoli. Such cases are hard to discern visually even for humans.*

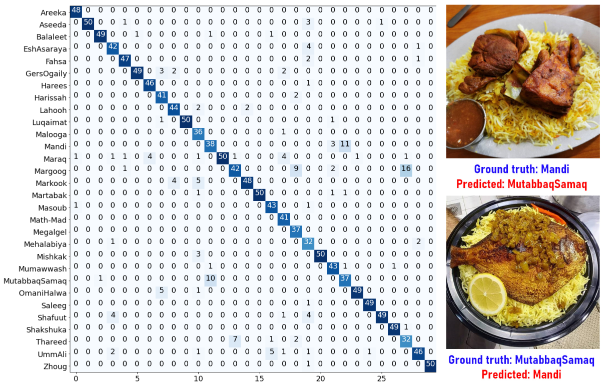


Figure 11: ***Fine-grained GCC-30 classification results.*** *Left: the confusion matrix shows that the model can discriminate all specialties. Right: some examples of incorrect classification on particularly complicated, rice-based dishes.*

application to web resources for providing complete information about the food specialties automatically recognized, and to compare this information to user-specific dietary requirements on the smart phone.

5. Conclusions and future work

We presented a food computing framework for end-to-end food image classification. Our framework spans the full range, from data set creation up to mobile deployment and usage in the wild. In particular, we demonstrate how our framework can be used for creating custom data sets representing regional gastronomy that is woefully underrepresented in presently available data sets. Our framework can support touristic, marketing, and political initiatives that aim at the promotion and protection of traditional and local food specialties. Using the Food101 data set as a baseline, we demonstrated that our process for training the classifiers exceeds state-of-the-art performance, reaching a top-1 accuracy of 91.91% using an EfficientNet-B4 We assessed our classification models with ETH's Food101 data set and obtained a top accuracy of 91.91% [TL19]). We also demonstrated the framework's versatility by creating two custom data sets and models for Pizza-Styles (including otherwise rather obscure, local variants such as the D.O.P. Focaccia di Recco from Liguria or Pizzetta Sfoglia from Sardinia), and for traditional dishes in Middle Eastern Gulf Countries (Yemen, Oman, Qatar, Saudi, Kuwait, etc.). Finally, we showed the potential of the system by taking our mobile app to the "isolated tourist in a real-world" setting: dining in traditional restaurants without relying on an active internet connection to classify food unfamiliar to some in our group. Despite promising preliminary outcomes, there are still important practical limitations in the framework that we plan to address in the future:

- **fully automatic pre-processing:** our data set creation step still needs manual proof-reading, a time consuming tedium. We plan to incorporate modern machine learning approaches for automating this process, in particular, by following curriculum learning strategies [BLCW09, WCZ21] and self-training pipelines [XLHL20]. Self training is exceptionally attractive in this context, since it offers a seamless transition path from supervised to unsupervised learning.
- **hierarchical classification:** we are currently able to create custom models representing particular food categories or regional cuisines. As long as more models will be created, we will need to create hierarchical organization of categories in a way to refine classification starting from food type, up to the region, and specific subcategories. To this end, we plan to employ hierar-

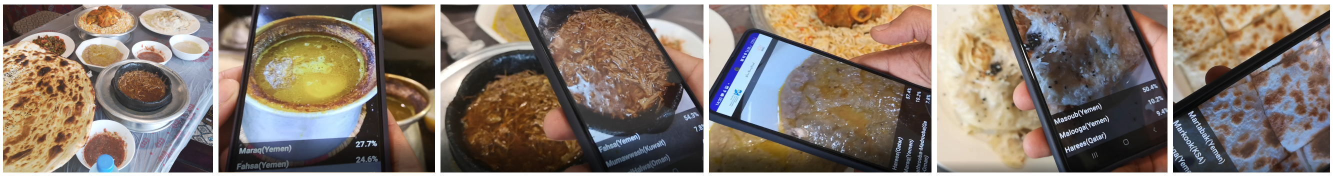


Figure 12: ***Real-time mobile application.*** *We tested our classification model trained on GCC-30 in a traditional Yemeni restaurant, resulting in successful recognition of numerous traditional dishes.*

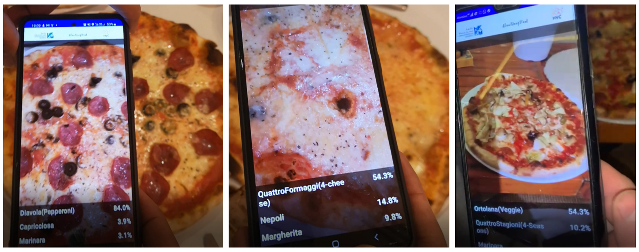


Figure 13: ***Real-time mobile application.*** *We tested our classification model trained on Pizza-Styles in an Italian Pizzeria, resulting in successful recognition of various pizza types.*

chically structured local classifiers [YWHL19]. In the same context, a largely unexplored area that deserves more attention in the future is the automatic extraction of taxonomies that are “beautiful to a machine”, rather than humans. While understandable taxonomies have established themselves as useful tools to categorize and organize knowledge for humans, they are not necessarily the right tool to make a hierarchical concept accessible to machine learning. We will therefore also examine the automatic generation of machine taxonomies, e.g., by hierarchical clustering based on confusion matrices.

- **few-shot learning:** according to the food specialties considered, it is very difficult to find online representative images to be included in the training sets for classification models. This, of course, is the curse of the premise of our work: We want to preserve variety and local influence over the internationally established gastronomic mean. Therefore, few-shot learning schemes [WYKN20] are needed to alleviate this issue, and we plan to investigate this avenue in the future.

Acknowledgments. The project was funded by the Information and Computing Technology department in the College of Science and Engineering at Hamad Bin Khalifa University. The color scheme for all diagrams of this paper was created by extracting dominant colors from the painting ’Bacchanal with Black Bull’ by artist Pablo Picasso.

References

[BGVG14] Bossard L., Guillaumin M., Van Gool L.: Food-101–mining discriminative components with random forests. In *European conf. on computer vision* (2014), Springer, pp. 446–461. 2, 3, 5, 6, 7

[BLCW09] Bengio Y., Louradour J., Collobert R., Weston J.: Curriculum learning. In *International Conference on Machine Learning* (2009), pp. 41–48. 9

[CMN20] Ciocca G., Micali G., Napoletano P.: State recognition of food images using deep features. *IEEE Access 8* (2020), 32003–32017. 2

[CNS15a] Ciocca G., Napoletano P., Schettini R.: Food recognition and leftover estimation for daily diet monitoring. In *Intern. Conf. on Image Analysis and Processing* (2015), Springer, pp. 334–341. 3

[CNS15b] Ciocca G., Napoletano P., Schettini R.: Food recognition and leftover estimation for daily diet monitoring. In *Intern. Conf. on Image Analysis and Processing* (2015), Springer, pp. 334–341. 3

[CZN*20] Chen J., Zhu B., Ngo C.-W., Chua T.-S., Jiang Y.-G.: A study of multi-task and region-wise deep learning for food ingredient recognition. *IEEE Transactions on Image Processing 30* (2020), 1514–1526. 2

[FAS14] Farinella G. M., Allegra D., Stanco F.: A benchmark dataset to study the representation of food images. In *European Conference on Computer Vision* (2014), Springer, pp. 584–599. 3

[FKMN21] Foret P., Kleiner A., Mobahi H., Neyshabur B.: Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations* (2021). URL: https://openreview.net/forum?id=6Tm1mposlrM. 7, 8

[GdMWP*20] Gonçalves D. N., de Moares Weber V. A., Pistori J. G. B., da Costa Gomes R., de Araujo A. V., Pereira M. F., Gonçalves W. N., Pistori H.: Carcass image segmentation using cnn-based methods. *Information Processing in Agriculture* (2020). 2

[HCB*19] Huang Y., Cheng Y., Bapna A., Firat O., Chen D., Chen M., Lee H., Ngiam J., Le Q. V., Wu Y., et al.: Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems 32* (2019), 103–112. 3

[HG20] Howard J., Gugger S.: Fastai: a layered api for deep learning. *Information 11*, 2 (2020), 108. 6, 7

[HMC*16] 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 2nd International Workshop on Multimedia Assisted Dietary Management* (2016), pp. 41–49. 8

[HSS18] Hu J., Shen L., Sun G.: Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 7132–7141. 3, 5

[JMLL20] Jiang S., Min W., Liu L., Luo Z.: Multi-scale multi-view deep feature aggregation for food recognition. *IEEE Transactions on Image Processing 29* (2020), 265–276. 2, 3

[JQL*20] Jiang L., Qiu B., Liu X., Huang C., Lin K.: Deepfood: Food image analysis and dietary assessment via deep model. *IEEE Access 8* (2020), 47477–47489. 2

[KSH12] Krizhevsky A., Sutskever I., Hinton G. E.: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012), 1097–1105. 3

[LAM*20] Latif G., Alsalem B., Mubarky W., Mohammad N., Alghazo J.: Automatic fruits calories estimation through convolutional neural networks. In *Proceedings of the 2020 6th International*

Conference on Computer and Technology Applications (2020), pp. 17–21. 2

[LCL*16] 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 *International Conference on Smart Homes and Health Telematics* (2016), Springer, pp. 37–48. 8

[LHL16] Li Y., Hao Z., Lei H.: Survey of convolutional neural network. *Journal of Computer Applications 36*, 9 (2016), 2508–2515. 3

[Liu20] Liu R.: Higher accuracy on vision models with efficientnet-lite. *TensorFlow Blog.[online] Available at: https://blog. tensorflow. org/2020/03/higher-accuracy-on-visionmodels-with-efficientnet-lite. html [Accessed 30 Apr. 2020]* (2020). 5

[LNC20] Lam M. B., Nguyen T.-H., Chung W.-Y.: Deep learning-based food quality estimation using radio frequency-powered sensor mote. *IEEE Access* (2020). 2

[LSV*20] Lu Y., Stathopoulou T., Vasiloglou M. F., Pinault L. F., Kiley C., Spanakis E. K., Mougiakakou S.: gofoodtm: An artificial intelligence system for dietary assessment. *Sensors 20*, 15 (2020), 4283. 3

[LTP*21] Li S., Tan M., Pang R., Li A., Cheng L., Le Q. V., Jouppi N. P.: Searching for fast model families on datacenter accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 8085–8095. 3

[MFM18] Martinel N., Foresti G. L., Micheloni C.: Wide-slice residual networks for food recognition. In *2018 IEEE Winter Conference on applications of computer vision (WACV)* (2018), IEEE, pp. 567–576. 3, 8

[MH21] Mercioni M. A., Holban S.: Soft-clipping swish: A novel activation function for deep learning. In *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (2021), IEEE, pp. 225–230. 5

[MJL*19] Min W., Jiang S., Liu L., Rui Y., Jain R.: A survey on food computing. *ACM Computing Surveys (CSUR) 52*, 5 (2019), #91:1–36. 1, 2

[MJR*15] Meyers A., Johnston N., Rathod V., Korattikara A., Gorban A., Silberman N., Guadarrama S., Papandreou G., Huang J., Murphy K. P.: Im2calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1233–1241. 3

[MLLJ19] Min W., Liu L., Luo Z., Jiang S.: Ingredient-guided cascaded multi-attention network for food recognition. In *Proceedings of the 27th ACM International Conference on Multimedia* (2019), pp. 1331–1339. 7, 8

[MLW*20] Min W., Liu L., Wang Z., Luo Z., Wei X., Wei X., Jiang S.: 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* (2020), pp. 393–401. 3

[MSFV*21] Medus L. D., Saban M., Francés-Víllora J. V., Bataller-Mompeán M., Rosado-Muñoz A.: Hyperspectral image classification using cnn: Application to industrial food packaging. *Food Control 125* (2021), 107962. 2

[OY21] Okamoto K., Yanai K.: Uec-foodpix complete: A large-scale food image segmentation dataset. In *International Conference on Pattern Recognition* (2021), Springer, pp. 647–659. 3

[Pop20] Poply P.: An instance segmentation approach to food calorie estimation using mask r-cnn. In *Proceedings of the 2020 3rd International Conference on Signal Processing and Machine Learning* (2020), pp. 73–78. 2

[QLS*19] Qiu J., Lo F. P. W., Sun Y., Wang S., Lo B.: Mining discriminative food regions for accurate food recognition. In *British Machine Vision Conference* (2019), British Machine Vision Association. 8

[QZC*18] Qin Z., Zhang Z., Chen X., Wang C., Peng Y.: Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (2018), IEEE, pp. 1363–1367. 3

[RKK18] Reddi S., Kale S., Kumar S.: On the convergence of adam and beyond. In *International Conference on Learning Representations* (2018). 7

[RMK20] Rachakonda L., Mohanty S. P., Kougianos E.: ilog: an intelligent device for automatic food intake monitoring and stress detection in the iomt. *IEEE Transactions on Consumer Electronics 66*, 2 (2020), 115–124. 3

[SDBJ21] Sarda E., Deshmukh P., Bhole S., Jadhav S.: Estimating food nutrients using region-based convolutional neural network. In *Proceedings of International Conference on Computational Intelligence and Data Engineering* (2021), Springer, pp. 435–444. 2

[SdSZ*18] Santos A. G., de Souza C. O., Zanchettin C., Macedo D., Oliveira A. L., Ludermir T.: Reducing squeezenet storage size with depthwise separable convolutions. In *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), IEEE, pp. 1–6. 3

[SGH*21] Sadler C. R., Grassby T., Hart K., Raats M., Sokolović M., Timotijevic L.: Processed food classification: Conceptualisation and challenges. *Trends in Food Science & Technology* (2021). 2

[SHZ*18] Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 4510–4520. 5

[Sim12] Simonetti L.: The ideology of slow food. *Journal of European Studies 42*, 2 (2012), 168–189. 2

[SLP*21] Srinivas A., Lin T.-Y., Parmar N., Shlens J., Abbeel P., Vaswani A.: Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 16519–16529. 3

[Smi18] Smith L. N.: A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820* (2018). 7

[SSC*20] Shen Z., Shehzad A., Chen S., Sun H., Liu J.: Machine learning based approach on food recognition and nutrition estimation. *Procedia Computer Science 174* (2020), 448–453. 2

[TL19] Tan M., Le Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (2019), PMLR, pp. 6105–6114. 3, 4, 5, 7, 8, 9

[WCZ21] Wang X., Chen Y., Zhu W.: A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 9

[WYKN20] Wang Y., Yao Q., Kwok J. T., Ni L. M.: Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR) 53*, 3 (2020), 1–34. 10

[XLHL20] Xie Q., Luong M.-T., Hovy E., Le Q. V.: Self-training with noisy student improves ImageNet classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 10687–10698. 9

[YWHL19] Yao H., Wei Y., Huang J., Li Z.: Hierarchically structured meta-learning. In *International Conference on Machine Learning* (2019), PMLR, pp. 7045–7054. 10

[ZS18] Zhang Z., Sabuncu M. R.: Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)* (2018). 7

[ZYK21] Zhao H., Yap K.-H., Kot A. C.: Fusion learning using semantics and graph convolutional network for visual food recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2021), pp. 1711–1720. 2

[ZZL*19] 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*, 6 (2019), 1793–1811. 2