Diabetes60 - Inferring Bread Units From Food Images Using Fully Convolutional Neural Networks

Patrick Ferdinand Christ*†[1], Sebastian Schlecht*[2], Florian Ettlinger[1], Felix Grün[1], Christoph Heinle[2], Sunil Tatavatry[2], Seyed-Ahmad Ahmadi[3], Klaus Diepold[2], and Bjoern H. Menze[1]

[1]Department for Computer Science, Technical University of Munich, Arccistrasse 21, 80333 Munich
[2]Department for Electric Engineering, Technical University of Munich, Arccistrasse 21, 80333 Munich
[3]Department for Neurology, University Hospital Grosshadern, Marchioninistrasse 15, 81377 Munich

Abstract

In this paper we propose a challenging new computer vision task of inferring Bread Units (BUs) from food images. Assessing nutritional information and nutrient volume from a meal is an important task for diabetes patients. At the moment, diabetes patients learn the assessment of BUs on a scale of one to ten, by learning correspondence of BU and meals from textbooks. We introduce a large scale data set of around 9k different RGB-D images of 60 western dishes acquired using a Microsoft Kinect v2 sensor. We recruited 20 diabetes patients to give expert assessments of BU values to each dish based on several images. For this task, we set a challenging baseline using state-of-the-art CNNs and evaluated it against the performance of human annotators. In our work we present a CNN architecture to infer the depth from RGB-only food images to be used in BU regression such that the pipeline can operate on RGB data only and compare its performance to RGB-D input data. We show that our inferred depth maps from RGB images can replace RGB-D input data at high significance for the BU regression task. In its best configuration, our proposed method achieves a $RMSE$ of 1.53 BUs using RGB and inferred depth. Considering the variability among the raters themselves of $RMSE = 0.89$, we can show that our baseline method with depth prediction can extract reasonable nutritional information from RGB image data only.

*Authors contributed equally
†Corresponding address: patrick.christ@tum.de

1. Introduction

1.1. Motivation

Diabetes mellitus is one of the most common chronic diseases worldwide and continues to increase from 285 million today to 439 million diseased people in 2030, as changing lifestyles lead to reduced physical activity, and increased obesity [34]. For diabetic patients an accurate caloric assessment of their nutritional intake is needed to regulate their dysfunctional blood sugar cycle. Diabetologists introduced a simplified scheme: the bread units or carbohydrate units to assess the nutritional intake of a meal. One bread unit corresponds to a quantity of food containing 12-15g of digestible i.e. blood-sugar-effective carbohydrates present in different forms of sugar or starch [39]. Diabetes patients learn the assessment of bread units (BU) by learning correspondence between BU and meals from textbooks and personal experience. Apart from experience, the process of estimating one's personal caloric intake may additionally require holistic knowledge about nutrition. Yet unknown dishes' BUs may be difficult to estimate, local customs in food preparation that are not visually apparent, e.g. preparing spaghetti with butter versus sunflower oil, may lead to additional uncertainty for experienced diabetes patients. Furthermore, there is a high uncertainty and danger of miscalculation for patients new to the disease. Digital support systems can be a way to provide guidance and help in those situations. Especially children could benefit significantly, due to their initially limited knowledge about their disease and nutritional values of food. Also, around 5% of pregnancies coincide with a short-term gestational diabetes mellitus (GDM) with potential harm for the unborn baby. With such a sudden onset GDM, affected pregnant women could also highly benefit from a computer aided diabetes assessment system [13]. Even though BU estima-

tion is a task that is very specific to diabetes, estimating the amount of carbohydrates and other micro nutrients is done in many more contexts like sports or weight-loss. A healthy diet is described not only by the kind of dish and its ingredients, but also by the amount which is consumed. In those cases, a digital system which processes meal images and derives rich information could provide additional support to reduce the effort of diets and better engage users in a healthy lifestyle. In this work we want to take a step towards computer aided nutrition assessment.

1.2. Related Work

Food Computer aided assessment of food and nutritional information of meals is an uprising research field in the computer vision community. Previous work can be categorized into meal classification, segmentation and caloric assessment. Public datasets so far focused on food classification such as Food101 [3], PFID [6], UNICT-FD889 [12], VIREO172 [5] and UECFOOD-100 [23]. Meyer et al. 2015 collected a 3D food dataset for assessing calories, but did neither publish their 3D data nor food classification data [25]. In the past, classical hand crafted features have been extracted to classify meals, ingredients or restaurant-specific multi-labels [3, 14]. Recently, deep convolutional neural network based methods are gaining also popularity in food classification [5, 21]. [7, 14, 25, 5] applied deep learning based segmentation methods to segment food on plates to perform higher level vision tasks. High level vision tasks include calory assessment [42, 25, 28], cooking recipe retrieval [5] and carbohydrate estimation [29]. Many of these approaches use structure from motion information from several images to develop a 3D food model [29, 7, 18].

Food volume estimation [41] and [4] used template based matching to estimate volumes of food. Especially [27] obtained very good results in regard to volume estimation using feature matching and pose estimation, however in order to obtain an absolute scale, a reference object was needed which had to be placed next to the food item.

Depth prediction Using RGB data as a basis to generate a corresponding depth image has been researched intensely whereas the classical approach in this field is merely using stereo-imagery. Scharstein et al. [33] for example investigated a broad range of existing algorithms based on stereo matching. These algorithms however rely on stereo cameras to work. More closely related to our experiments are methods trying to generate depth information from more loosely aligned images. Sturm et al. [38] presented an effective way to obtain proper scaling for consecutive images in order to calculate 3D structure and motion information - the algorithm thus relies on a sequence of consecutive images. In a more unconstrained setting, Snavely et al. were [37] using many unstructured images from popular sites to generate a 3D view. The underlying system in this case is also based on features and keypoints which are later matched. Machine learning itself has also already been applied to stereo imagery and depth estimation as shown in [17]. Also, in [24] deep neural networks have been trained to be able to predict disparity by learning binocular filters. These systems could then be used as support for stereo setups. Very closely related to depth prediction from single still images are the works from Eigen et al. [11][10], Laina et al. [20] or Liu et al. [22] who use neural networks to infer the depth of still images.

1.3. Contribution

Our contributions in this work are fourfold.

First, we provide and formulate a new computer vision task of inferring bread units (BU) from RGB or RGB-D data by publishing 9k RGB-D image pairs rated by 20 experts.

Second, we present an automatic method for BU regression given RGB-D images using residual neural networks.

Third, we propose a new fully convolutional neural network architecture using skip connections to infer depth maps from RGB images. The architecture at hand shows very good convergence behavior and is especially suited for prediction tasks where small relative errors and local details are especially important.

Finally, we present an automatic method of regressing BUs given only RGB images in two steps by 1) predicting the depth map from the RGB image and 2) predicting the BUs from the RGB image and the predicted depth map.

2. Dataset

2.1. Data Acquisition

Our hardware setup for data-collection consists of a Microsoft Kinect v2 sensor which is connected to a laptop via USB. Since the original Kinect v2 is primarily powered by a 230V power supply, its portability is rather limited. To overcome this issue, we connected the device to a 12V battery-pack making it suitable for mobile use. The device captures depth in a 512x424 pixel frame by default while providing a 1920x1080 pixel RGB output [1]. We collected a total of about 9k RGB-D pairs of 60 different western dishes. The image-streams have been recorded from various angles and distances to capture a wide range of perspectives for each dish. Even though version 2 of the Kinect sensor improved in terms of available ranges, it is still required to maintain a certain minimum distance to the object of interest to receive valid depth values from the device. During recording, we projected the incoming depth stream onto the RGB frame, thus we only provide the projected depth-map in our dataset. Since valid depth is also only provided within certain parts of the RGB frame's spatial dimensions due to the

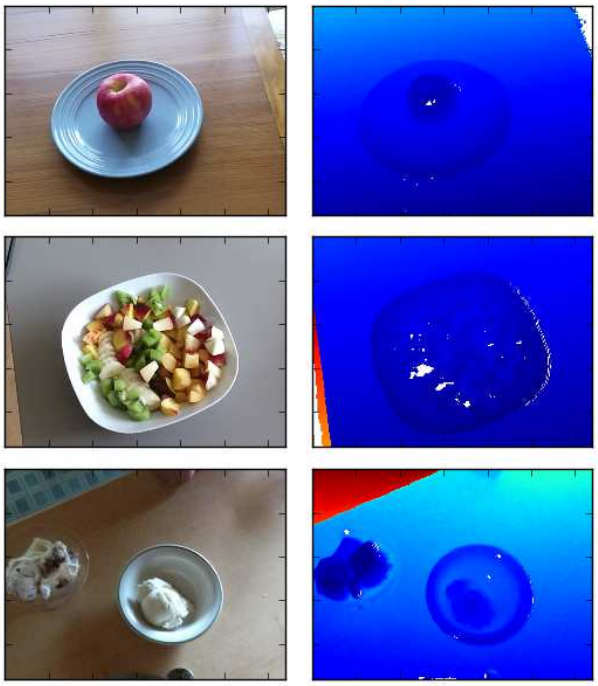

Figure 1. RGB frame (left) and registered depth frame (right) of exemplary classes *Apple*, *Fruit Salad* and *Ice cream*. This figure is best viewed in color.

smaller size of the obtained depth frame, we center crop the RGB-D pair at a size of 640x480 pixel. Since we tried to keep operating the device within a certain maximum distance, the majority of the depth measurements are between 60-80cm. In some scenes, background structures such as floors, chairs or adjacent rooms are visible. Since those pixels exhibit a depth with is mostly larger than 1.2m, they can easily be masked if necessary. Similar to [35] and [26] we experienced similar artifacts degrading the quality of the depths maps such as occlusions from specular or low albedo surfaces, as well as shadowing caused by the physical alignment of infrared emitter and camera. Especially plates, glasses, cutlery or greasy food show a frequent absence of valid depth measurements. Since the algorithm presented in section 3.2.1 has a natural ability to deal with missing depth values by neglecting them during cost computation, we did not see the necessity to fill in missing values during post-processing. From the incoming stream of data, we dumped equally spaced RGB-D pairs at a frequency of about 8-10 frames per second. Due to buffering inconsistencies with the underlying library that we used to interface the Kinect, the capture frequency may differ slightly from recording to recording. The dataset may also contain some slight noise such as blurs from camera movement or partial occlusion through other objects due to the fact that is has been recorded by a handheld device without a tripod. However, we removed unusable frames from the data. Exemplary recordings of the dataset can be seen in figure 1.

2.2. Dataset Specifics

Our dataset compromises 60 western dishes with RGB images and depth-maps (RGB-D) with a total of 8820 images, i.e. 147 images per dish on average. The 60 western dishes were chosen in such a way to cover common meal types. The dataset contains dishes from various categories like "Salads", "Traditional" or "Breakfast". The dishes have been recorded at various locations around TUM university campus, cafeteria or at home. The distribution of these categories in the dataset can be seen in figure 2.

To learn the correspondence from RGB-D to BU, we surveyed 20 long-term diabetic melitus type 1 patients to estimate the bread unit count for our 60 dishes. We showed them a RGB image of a meal and asked them to estimate the bread units. The assessment has been conducted via a proprietary web-application to which images could be uploaded and presented to annotators by sending them a link to the application. Each annotator could then browse through each of the images individually and assign a single BU value per dish. We set the maximum precision per rating to 0.5 BU.

Figure 3 shows the boxplot of the expert BU ratings. The average BU of our data is 3.49 and the averaged BU STD is 0.89 with a minimum STD of 0 BUs (all raters rated the same value) and maximum of STD of 1.99 BU where rater opinions highly disagreed.

3. BU Prediction

Since the assessment of bread units (BU) strongly depends on the volume of the food, we took the depth information of our food dataset into account. We state this problem in the following way:

$$BU(V, \rho_{Food}) = V \cdot \rho_{Food} \tag{1}$$

With the volume V of the meal and ρ_{Food} the bread unit density. The volume V of a meal can be stated as:

$$V \approx \iint_{\mathcal{F}} h d\sigma \tag{2}$$

Where h is the measured depth value of the meal taken from top view normalised such that depth values are zero outside the dish and $\mathcal{F}$ is the projected area of the dish. I.e. we make two assumptions: a) dishes do not overhang b) dishes have a homogeneous bread unit density.

We present two experiments to regress to bread units from our dataset. In all experiments we train on the images of 40 dishes and test on the other 20 dishes of our dataset, i.e. we evaluate our networks capability of predicting BU of categories of food it has never seen before. We use 3-fold cross-validation.

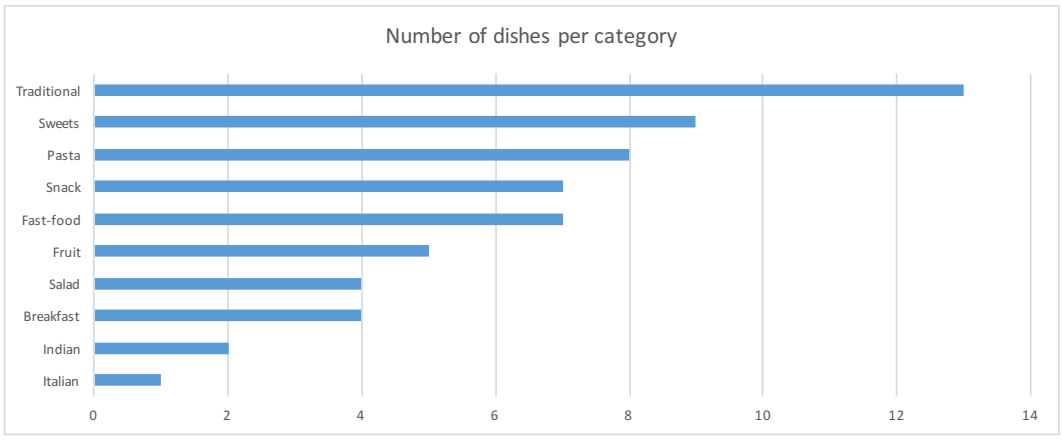

Figure 2. Distribution of different food categories present in the Diabetes60 dataset.

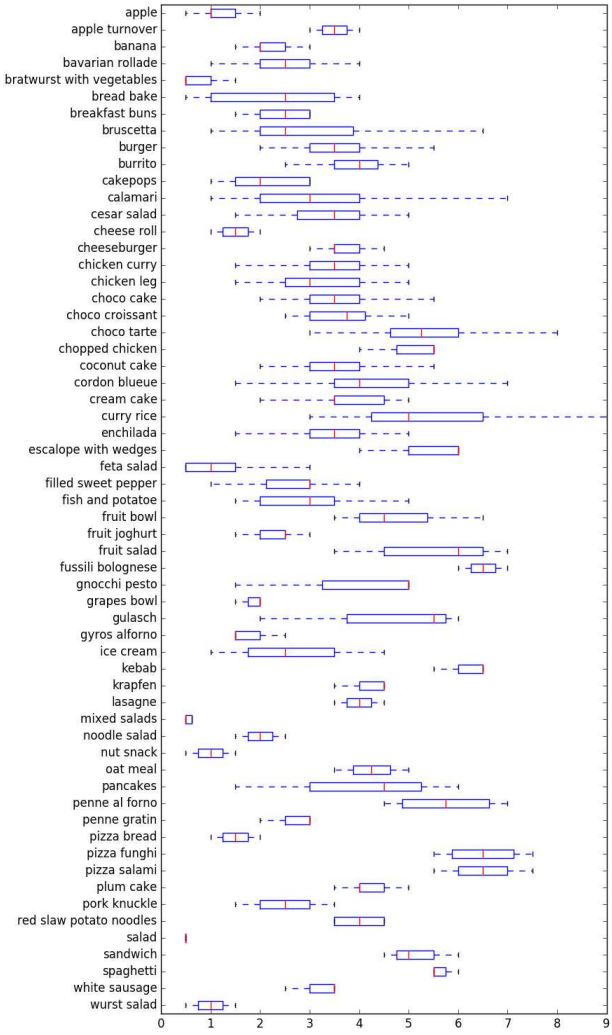

Figure 3. Boxplots of the Bread Unit (BU) estimates from diabetic patients for each class id.

In our first experiment we regress the bread units given the RGB and the corresponding ground-truth depth map obtained from the Kinect using a state-of-the-art Convolutional Neural Network architecture pretrained on the Food 101 dataset. The architecture of choice is Resnet-50 as proposed in [15]. We selected this type of data for pre-training since the task domains are similar. In both cases, images of foods are used for input. To make the network regress values instead of producing a certain class probability, we changed the cross-entropy loss to $\mathcal{L}_2$. In the second experiment, we trained a fully convolutional neural network to predict the depth map of a given RGB image to remove the necessity to have a depth camera. We fine-tuned the depth prediction model on top of the NYU Depth v2 dataset [26]. Afterwards, we trained the Resnet-50 with the predicted depth maps produced by the trained depth predictor. During test time we only provided RGB to regress the bread units. To obtain ground-truth values for the bread units, we averaged the individual ratings per dish. An overview of our conducted experiments is shown in figure 4. All our experiments were conducted on an Ubuntu workstation equipped with a 12GB NVIDIA TITAN X GPU. The neural networks were assembled using the Deep Learning framework Lasagne[8]. In our setup, we downsample all frames by a factor of 2 within the dataset, yielding spatial dimensions of 320x240. For our networks' inputs, we chose images of 304x228 in size, such that there is space for random cropping to further augment the data. In addition to random cropping we use random horizontal flips for augmentation. We also normalize all inputs $x_{i,c}$ with c being the 4 channels via simple precomputed statistics as seen in equation 3

$$x^*_{i,c} = \frac{x_{i,c} - \mu_{i,c}}{\sigma_c} \tag{3}$$

with $\mu_{i,c}$ being the pixel- and channel-wise mean for each pixel i and each channel c. σ_c denotes the standard-

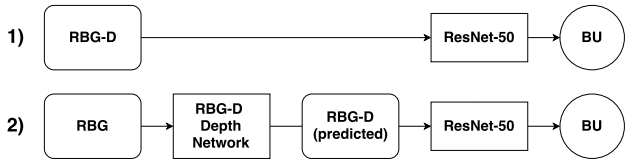

Figure 4. Pipelines of the conducted experiments. 1) BU prediction from RGB-D images, 2) BU prediction from RGB images with intermediate RGB-D prediction using the RGB-D Depth Network.

deviation for each channel c, both computed for the dataset we use for training.

3.1. BU Prediction from RGB and measured depth

We model bread unit estimation by using the depth information as an additional channel to our CNN architecture.

In this experiment, we used a pre-trained the Resnet-50 model on RGB data but tried to preserve the filter learned on the color input channels. We thus initialized that part of the weight tensor corresponding to the RGB input with the weights from the pre-trained network, whereas the part of the filters operating on the depth input channel was initialized with random values following the initialization scheme in [16].

In order to train the residual network for direct bread-unit regression, we replaced the last softmax-layer with a single neuron (ReLU activation) and corresponding L_2 loss. Initial experiments on training the network from scratch led to bad convergence behaviour and overall bad performance. Instead, initializing the weights of neural networks from related tasks often not only promotes convergence but can also lead to higher absolute performance [40]. Therefore, we pre-trained the Resnet-50 model to classify food images first. The Food101 dataset [3] features around 101k images of western foods of various categories. The network achieved a top-1 accuracy of around 70% ([3]: 50.7%). In all BU regression experiments, we used a starting learning rate of 10^{-3} and trained the network for 40 epochs using SGD with momentum and reduced the learning rate once we observed plateaus. Momentum was set to 0.95, whereas we used a weight-decay factor of 10^{-4}.

3.2. BU Prediction from RGB and Inferred Depth

The availability of depth information can provide an additional channel to derive features from, its availability is often lower compared to RGB data. Even though there are handheld devices such as Google's Tango [2] that allow for mobile depth perception, the vast majority of todays' mobile devices are solely equipped with a single RGB camera. This motivates the use of a model to predict the corresponding depth map to a given input image such that only a single RGB image is required to regress the amount of bread units for a given dish.

3.2.1 Inferring Depth from RGB

In a real-life scenario, a diabetes patient is more likely to have access to a camera equipped device such as a smartphone compared to a device equipped with a structured light sensor or a stereo camera setup. We thus want to incorporate a model into our pipeline that estimates the depth of a given scene using only RGB data from a single image. However, mapping from RGB input values to depth is a physically ill-posed problem and with only a single image, this ambiguity cannot be removed. In practice, it is however possible to find a model that can predict depth with reasonable accuracy. The reason is the fact that, apart from unlikely extremes, objects often tend to have similar dimensions in the context of particular scenes, rendering neural networks capable of finding good generalizations to map from an image to its corresponding depth.

Architecture Several works like [10] [11] [32] [22] [20] have already addressed this issue using Deep Neural Networks. In our work, we used an architecture closely related to the one proposed in [20] and performed a set alterations. This architecture has proven to be superior to architectures based on convolutions and fully connected layers such as AlexNet- [19] or VGG-based networks [36] because it is solely composed of convolutional layers while still being able to obtain a receptive field large enough to grasp the whole scene. We made small changes however by incorporating skip connections as proposed in [30]. The purpose of those connections is to provide features of small scales to the later expansive path of the network to preserve local details of the food items. In addition, works like [9] have shown that this approach can ease training and improve overall results. We observed low convergence rates when training an architecture without skip connections completely from scratch as proposed in [20]. With skip connections however, the model converged reliably fast, which allowed end-to-end training in all training cases. To implement that, we also altered the expanding path such that the spatial dimensions of the feature maps match those in the contracting path. This allows for concatenating the activations without cropping. The overall architecture for the depth prediction model is shown in figure 5. Please note that in convolution (symbol: *), no reduction of spatial dimensions takes place. Each orange-colored arrow represents a sequence of residual blocks, the length of the sequence is depicted by the number on the left side of each arrow. The first block in the sequence does always have a shortcut with a projection convolution in place. As in Resnet-50, we use bottleneck blocks. See [15] for details. Our upsampling blocks are conventional residual blocks with projection shortcuts that receive up-scaled versions of the previous feature map with a scaling factor of 2 and use convolutional

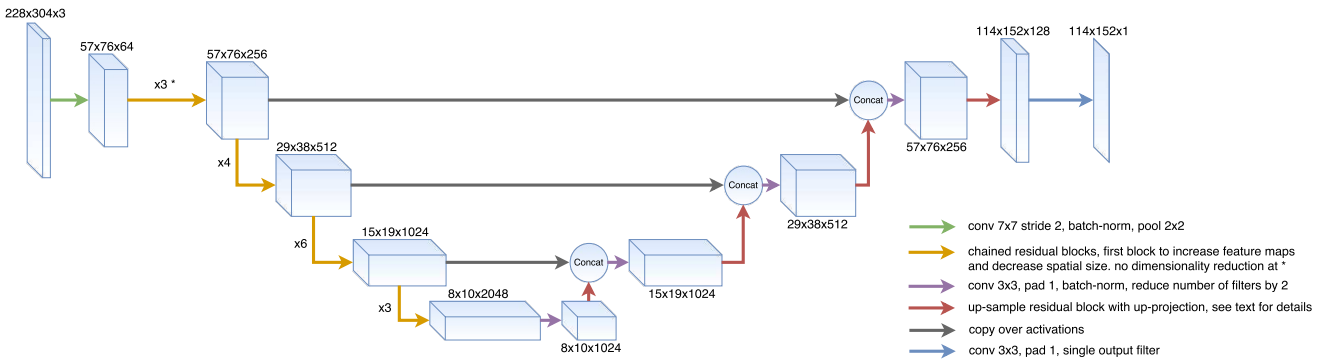

Figure 5. RGB-D Depth Network Architecture with skip connections: The network has an encoding and decoding pathway. Skip connections introduced by [30] allow spatial information exchange and promote convergence.

filters in sizes from 4x4 to 5x5 such that the spatial dimensions of the output feature maps match those obtained in the corresponding contractive counterpart to ease concatenation.

Loss Function We use the reverse-Huber loss function for depth prediction as introduced in [20]. This function is expressed in equation 4 with $d = D - D^*$, where D and D^* denote the prediction and ground-truth depth maps:

$$\mathcal{B}(d) = \begin{cases} |d|, & |d| \leq c \\ \frac{x^2+c^2}{2c}, & |d| > c \end{cases} \tag{4}$$

and c being $\frac{1}{5} max_i(|d_i|)$ for the pixel-wise residuals d_i.

Pre-training We pre-trained the model on the NYU Depth v2 dataset for indoor scene segmentation [35] to start off with a better weight configuration. In contrast to [20], [10] or [11] we did not initialize the contractive part of the network with weights. We leave two-staged pre-training open for future work (training the contractive part on a classification task first, then finetuning on NYU Depth v2, then finetuning on the target dataset).
We extracted equally spaced frames from the raw dataset to obtain a total number of around 26k RGB-D pairs which we globally shuffled afterwards. To actually verify whether the network generalizes, we used the official train/test split of the dataset. To make the data fit the network's inputs, we downsampled the frames by a factor of two using nearest-neighbour interpolation. It is important not to use a higher order interpolation method as they tend to interpolate between valid and invalid pixels. Augmentation was performed on-the-fly during training. The following methods were used to augment the data following values in [10]:

- **Random rotation** Rotating image and ground-truth in-plane for a random angle $\alpha \in [-5, 5]$
- **Zooming** Zooming the image and randomly select a part of it. The zooming factor was drawn per image within a range of [1, 1.5].
- **Random cropping** Similar to [19] we randomly crop images and ground-truth toward the desired network input size
- **Horizontal flips** Images and ground-truth are flipped horizontally with a probability of $p = 0.5$.
- **Random RGB scaling** Input images are randomly scaled with a pixel value $\beta \in [0.9, 1.1]^3$
- **Exposure** We made small changes in exposure to simulate various lighting conditions for the RGB input.

For pre-training we used a starting learning rate of 10^{-2}. In total, we extracted only abound 26k frames from NYU Depth v2 on which we trained the network for 80 epochs using SGD with momentum. We decreased the learning rate following a step-wise policy with a step-width of 20 epochs. The learning rate was decreased by a factor of $\gamma = 0.5$ per step.

Fine-tuning We fine-tuned our network on the data we collected. For training we split the 60 scenes into a training and test set using a split-factor of 0.75 resulting in about 7k frames for training and around 2k frames for test. We made sure that all frames belonging to a certain dish would end up either in the training set or in the test set. To further augment the data, we used the same processing pipeline as for the pre-training step. We trained the network for 40 epochs following a step-wise policy, starting with a learning rate of 10^{-3}, a step-width of 20 epochs while reducing the learning rate by a factor of $\gamma = 0.1$. Additionally, we mask out values larger than 1.2m, since those distances primarily belong to backgrounds in the image. Our experiments revealed also that the inclusion of masks yielded smoother gradients in the estimated depth maps.

	RMSE (lin)	RMSE (log)	rel	$\delta_1 = 1.25$	$\delta_2 = 1.25^2$	$\delta_3 = 1.25^3$
Our network	0.119	0.161	0.129	0.781	0.995	0.999

Table 1. Quantitative results for depth regression on 3D food data. For training and test we masked out all depth values larger than 1.2m in order to ignore the surfaces belonging to background.

4. Results

4.1. Depth Prediction

Our proposed depth architecture achieves state-of-the-art RMSE of 0.651m ([10]: 0.753m and 0.641m, [11]: 0.877m) on NYU Depth v2 when training completely from scratch. Since we were primarily interested in using those weights as a starting point for regressing the depth of food images, we did not extensively fine-tune the hyper parameters for this learning task or pre-trained the contractive stem on a large dataset like Imagenet [31].

Qualitative results of our model trained on the newly recorded food data are shown in figure 6. The results show that the model is able to grasp fine, local details, as seen in the example of the peaches in the bowl of fruits. We hypothesise that the skip-connections not only helped to make the model converge when pre-training, but also support to predict local structures, especially, when looking at the overall range of values. Local food structure is mostly in the range of only 1-2 centimeters whereas the overall depth ranges from about 60cm to partially up to over 1.2m. This is a result of the data being recorded handheld. A similar refinement effect has been reported by Eigen et al. in [11, 10] by using refinement stages in later stages of the network to improve the prediction. In contrast to [25], the depth maps for for food images predicted by our model feature fine details. Even though their model also has refinement stages, the resulting depth maps from our model are with spatial dimensions of 152x114 fairly large.

Furthermore, by masking out invalid depth values during loss-computation, the model also becomes inherently robust to deal with missing/invalid pixel-data from the Kinect. This makes inpainting or other filling techniques unnecessary, even though data recorded with the Kinect v2 is already less prone to contain large amounts of invalid pixels compared to earlier versions of the device. Quantitative metrics for our dataset are shown in table 1. The linear RMSE is 0.119 with a relative error of 0.129. These metrics set the baseline for the depth prediction task of our new dataset.

4.2. BU Prediction

Table 2 shows the results of the BU prediction. The CNN trained on RBG-D data yields an RMSE of 1.46. Trained end-to-end just on RBG with inferred depth achieved a RMSE of 1.53. Those results were obtained using 3-fold cross-validation. Figure 7 shows the box plot of predictions from RBG with inferred depth for all 60 dishes in our dataset. Our methods achieves for many dishes reasonable predictions within the spread of human expert ratings.

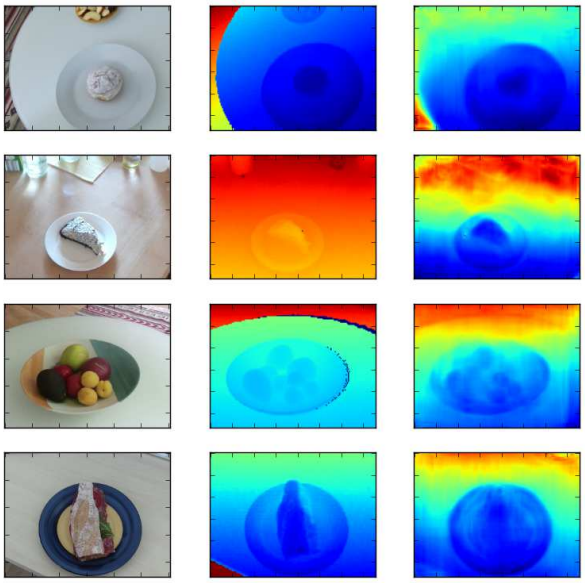

Figure 6. Qualitative results of the model on dishes of the categories *snack*, *sweets* and *fruit*. RGB input (left), ground-truth depth (middle) and predicted depth (right) are shown. The dishes shown above are part of our test-set, images are scaled individually. This figure is best viewed in color.

Approach	Root Mean Square Error (RMSE)
RGB-D Ground truth	**1.46**
RGB-D Predicted Depth	1.53

Table 2. Bread unit inference using Convolutional Neural Networks and Fully Convolutional Neural Networks.

When training the CNN to predict bread-units, the network converged quite quickly as very common in fine-tuning scenarios. This still holds when we use 4 input channels instead of 3 as we preserve the filters operating on color input by transplanting the weights. Providing predicted depth expectedly yields worse results compared to ground-truth depth input even though the margin of error is relatively small. The high accuracy of the predicted depth maps helped to obtain very close results. To further investigate this relation we calculated a Wilcoxon signed-rank test to determine whether RGB-D with ground truth or prediction do lead to the same RMSE. We found that the two approaches do produce the same output distribution with a p-value of 1.52×10^{-12}. We can conclude, that our method with predicted depth does convey the same results.

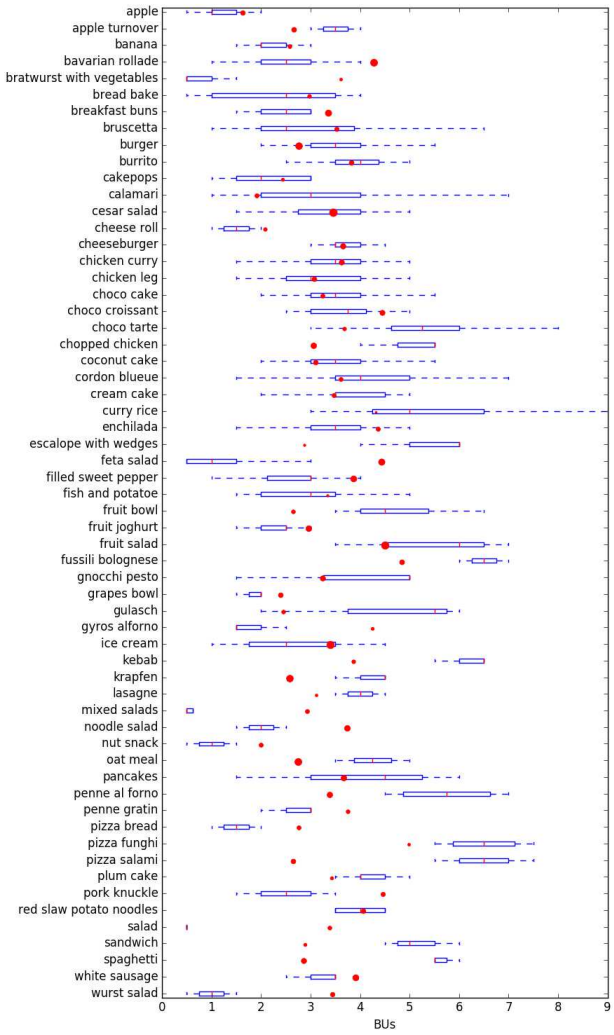

Figure 7. Box plot of ratings and the predictions given by the CNN using RGB and inferred depth. The BU ground truth by the expert annotators is shown in blue boxes and the average predicted BU in shown as a red dot.

5. Discussion and Conclusion

In this work we proposed a new computer vision task of inferring bread units from food image data. We collected RGB-D images of 60 western dishes and surveyed 20 experts to assess the bread unit count of the dishes. We demonstrated two methods of inferring bread units from RGB-D and RGB with a inferred depth map of a fully convolutional depth regression network. The high inter-rater RMSE of 0.89 shows that the task at hand is in fact very hard to solve, even to long-term Diabetes patients. Compared to human raters, our implemented methods perform automatic BU estimation at a RMSE of 1.53 for RGB + inferred depth, which sets a baseline for this task on our proposed dataset. For most dishes, our method yields reasonable BU estimates, i.e. within the standard deviation of human expert raters. However, there are also dishes with faulty BU inference outside this range, in particular pizza salami, spaghetti and salads. This highlights the challenging nature of our proposed learning task for state of the art computer vision methods.

We tried to accomplish a similar goal as [25, 4, 37, 29] in an end-to-end fashion. Calorie and bread unit assessment are closely related tasks and both rely on depth or volume of a meal, besides contextual and semantic information. We addressed the contextual and semantic information using state of the art residual neural network architectures as proposed by [15]. In our approach, to incorporate the depth and volumetric information, we neither relied on structure from motion information such as [29, 7, 18] nor on a reference object [27]. State of the art depth network architectures as proposed by [11, 20] did not converge on our dataset without pre-training. The relative error of our proposed depth network architecture on Diabetes60 is 0.129. Unfortunately the food depth data of [25] was not published. They reported a relative error of 0.18 on their food data [25]. Comparing their qualitative depth predictions (see figure 6c in [25]), our RBG depth network could reconstruct finer details of the food as shown in figure 6. Our proposed depth prediction architecture may also be useful to other high-dimensional regression tasks where pretrained weights are not available or there is a strong focus on local details. We hope that by publishing our dataset along with baseline methods and results, we provide a starting point for researchers to tackle the same or comparable problems, either in a similar end-to-end fashion or by splitting the task into several sub-tasks and solving them independently. The depth values at hand may also be useful for people working on 3D reconstruction and modeling of food items which may or may not be part of a pipeline achieving a different end goal. Public RBG-D datasets are rare and we hope to foster computer vision research in this field with our dataset contribution. Advancements in the fields of automated assessment of food intake could become highly valuable for diabetes patients or generally everyone keen on keeping track of her or his nutrition. Right now, all our models require fairly recent desktop GPUs to operate. Once deep learning becomes more adopted by smartphones or other portable devices, those models could operate on device and thus provide faster feedback. In addition, location services could be integrated tightly into the estimation process to leverage local information obtained from restaurants or food courts.

6. Acknowledgements

This work was supported by the Technical University of Munich - Institute for Advanced Study (funded by the German Excellence Initiative and the European Union Seventh

Framework Program under grant agreement n 291763), the Marie Curie COFUND program of the the European Union (Rudolf Mossbauer Tenure - Track Professorship to BHM) and the BMBF Softwarecampus project by Patrick Christ. We thank NVIDIA and Amazon AWS for granting GPU and computation support.

References

[1] Kinect V2. https://developer.microsoft.com/en-us/windows/kinect/hardware. Accessed: 2016-05-22. 2

[2] Tango, Google Developers. Accessed: 2016-05-22. 5

[3] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 2, 5

[4] J. Chae, I. Woo, S. Kim, R. Maciejewski, F. Zhu, E. J. Delp, C. J. Boushey, and D. S. Ebert. Volume estimation using food specific shape templates in mobile image-based dietary assessment. In *IS&T/SPIE Electronic Imaging*, pages 78730K–78730K. International Society for Optics and Photonics, 2011. 2, 8

[5] J. Chen and C.-W. Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 32–41. ACM, 2016. 2

[6] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In *ICIP*, pages 289–292. IEEE, 2009. 2

[7] J. Dehais, S. Shevchik, P. Diem, and S. G. Mougiakakou. Food volume computation for self dietary assessment applications. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pages 1–4. IEEE, 2013. 2, 8

[8] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, D. M. de Almeida, B. McFee, H. Weideman, G. Takcs, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French, and J. Degrave. Lasagne: First release., Aug. 2015. 4

[9] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The importance of skip connections in biomedical image segmentation. *CoRR*, abs/1608.04117, 2016. 5

[10] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *CVPR*, pages 2650–2658, 2015. 2, 5, 6, 7

[11] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014. 2, 5, 6, 7, 8

[12] G. M. Farinella, D. Allegra, and F. Stanco. A benchmark dataset to study the representation of food images. In *ECCV*, pages 584–599. Springer, 2014. 2

[13] A. Ferrara. Increasing prevalence of gestational diabetes mellitus a public health perspective. *Diabetes care*, 30(Supplement 2):S141–S146, 2007. 1

[14] H. He, F. Kong, and J. Tan. Dietcam: Multiview food recognition using a multikernel svm. *IEEE Journal of biomedical and health informatics*, 20(3):848–855, 2016. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4, 5, 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 5

[17] K. Konda and R. Memisevic. Unsupervised learning of depth and motion. *arXiv preprint arXiv:1312.3429*, 2013. 2

[18] F. Kong and J. Tan. Dietcam: Automatic dietary assessment with mobile camera phones. *Pervasive and Mobile Computing*, 8(1):147–163, 2012. 2, 8

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 5, 6

[20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*, abs/1606.00373, 2016. 2, 5, 6, 8

[21] 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 *International Conference on Smart Homes and Health Telematics*, pages 37–48. Springer, 2016. 2

[22] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, pages 5162–5170, 2015. 2, 5

[23] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *International Conference on Multimedia and Expo*, pages 25–30. IEEE, 2012. 2

[24] R. Memisevic and C. Conrad. Stereopsis via deep learning. In *NIPS Workshop on Deep Learning*, volume 1, 2011. 2

[25] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: towards an automated mobile vision food diary. In *ICCV*, pages 1233–1241, 2015. 2, 7, 8

[26] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3, 4

[27] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney. Recognition and volume estimation of food intake using a mobile device. In *WACV*, pages 1–8. IEEE, 2009. 2, 8

[28] D. Rav, B. Lo, and G. Z. Yang. Real-time food intake classification and energy expenditure estimation on a mobile device. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6, June 2015. 2

[29] D. Rhyner, H. Loher, J. Dehais, M. Anthimopoulos, S. Shevchik, R. H. Botwey, D. Duke, C. Stettler, P. Diem, and S. Mougiakakou. Carbohydrate estimation by a mobile phone-based system versus self-estimations of individuals with type 1 diabetes mellitus: A comparative study. *Journal of medical Internet research*, 18(5), 2016. 2, 8

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 5, 6

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 7

[32] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31(5):824–840, 2009. 5

[33] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 2

[34] J. E. Shaw, R. A. Sicree, and P. Z. Zimmet. Global estimates of the prevalence of diabetes for 2010 and 2030. *Diabetes research and clinical practice*, 87(1):4–14, 2010. 1

[35] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011. 3, 6

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[37] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 2, 8

[38] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, pages 709–720. Springer, 1996. 2

[39] H. Warshaw and K. Kulkarni. *Complete Guide to Carb Counting: How to Take the Mystery Out of Carb Counting and Improve Your Blood Glucose Control*. American Diabetes Association, 2011. 1

[40] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014. 5

[41] Y. Yue, W. Jia, J. D. Fernstrom, R. J. Sclabassi, M. H. Fernstrom, N. Yao, and M. Sun. Food volume estimation using a circular reference in image-based dietary studies. In *Proceedings of the 2010 IEEE 36th Annual Northeast Bioengineering Conference (NEBEC)*, pages 1–2. IEEE, 2010. 2

[42] W. Zhang, Q. Yu, B. Siddiquie, A. Divakaran, and H. Sawhney. snap-n-eat food recognition and nutrition estimation on a smartphone. *Journal of diabetes science and technology*, 9(3):525–533, 2015. 2