Accepted Manuscript

Combining deep residual network features with supervised machine learning algorithms to classify diverse food image datasets

Patrick McAllister, Huiru Zheng, Raymond Bond, Anne Moorhead

PII: S0010-4825(18)30038-6

DOI: 10.1016/j.compbiomed.2018.02.008

Reference: CBM 2894

To appear in: *Computers in Biology and Medicine*

Received Date: 25 May 2017

Revised Date: 11 February 2018

Accepted Date: 12 February 2018

Please cite this article as: P. McAllister, H. Zheng, R. Bond, A. Moorhead, Combining deep residual network features with supervised machine learning algorithms to classify diverse food image datasets, *Computers in Biology and Medicine* (2018), doi: 10.1016/j.compbiomed.2018.02.008.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Combining Deep Residual Network Features with Supervised Machine Learning Algorithms to Classify Diverse Food Image Datasets

Patrick McAllister[1], Huiru Zheng[1*], Raymond Bond[1], Anne Moorhead[2] [1]Ulster University,
[1]School of Computing; {mcallister-p2, *h.zheng, r.bond} @ulster.ac.uk
[2]Ulster University, School of Communication and Media; a.moorhead@ulster.ac.uk

Abstract

Obesity is increasing worldwide and can cause many chronic conditions such as type-2 diabetes, heart disease, sleep apnea, and some cancers. Monitoring dietary intake through food logging is a key method to maintain a healthy lifestyle to prevent and manage obesity. Computer vision methods have been applied to food logging to automate image classification for monitoring dietary intake. In this work we applied pretrained ResNet-152 and GoogleNet convolutional neural networks (CNNs) to extract features from food image datasets; Food 5K, Food-11, RawFooT-DB, and Food-101. Deep features were extracted from CNNs and used to train machine learning classifiers including artificial neural network(ANN), support vector machine(SVM), Random Forest, fully connected Neural Networks, and Naive Bayes. Results show that using ResNet-152 deep features with SVM with RBF kernel can accurately detect food items with 99.4% accuracy using Food-5K food image dataset. Trained with ResNet-152 features, ANN can achieve 91.34%, 99.28% when applied to Food-11 and RawFooT-DB food image datasets respectively and SVM with RBF kernel can achieve 64.98% with Food-101 image dataset. From this research it is clear that using deep CNN features can be used efficiently for diverse food item image classification. The work presented in this research shows that pretrained ResNet-152 features provide sufficient generalisation power when applied to a range of food image classification tasks.

Keywords: obesity, food logging, deep learning, convolutional neural networks, feature extraction

1. Introduction

Obesity is a global concern and is a serious health condition that can cause diseases such as heart disease, type-2 diabetes, and some cancers [1]. The in- crease of obesity has also been reported as a major burden on health care institutions through direct and indirect costs [56]. One of the major ways that obesity can be managed is through dietary management methods such as food logging and other methods [3]. Food logging is an activity in which the user document their energy intake to monitor their diet. Other methods may include the use of an exercise log book to document physical activities and the duration. Previously, users documented their intake using a food diary however many users now use smartphone applications to document their energy intake. The increase in smartphone usage has also led to the increase of well-being ap- plications that are able to facilitate food logging. Many of these applications incorporate a simple diary

ACCEPTED MANUSCRIPT

entry, and/or connect to an online database/API to search for nutritional content for each of the users entries. Other novel methods include allowing the user to photograph the food items to determine calorie values. Using images has the potential to remove much of the complexity from traditional food logging to make it convenient for the user to document food intake to promote dietary management. Many studies have been completed in researching the use of computer vision methods to classify photographs of food to promote food logging [4-6]. This interactive approach to food logging using the camera within a smart-device may promote the use of food logging which is an important method to maintain weight loss. The remainder of this paper is structured as follows: Section 2 presents related work in how this problem has been tackled in previous research. Section 3 discusses the aim, objectives, and contributions of this work. Section 4 describes the methods used in this work and the use of Convolutional Neural Networks (CNNs) for feature extraction. Experiment results are presented in Section 5 followed by a discussion in Section 6. Section 7 highlights study limitations and areas for future work.

2. Related Work

Food logging is a beneficial method to aid dietary management and recent novel methods have utilised meal photographs for food logging. A review [41] was completed to highlight a variety of computer vision methods that have been applied in food image recognition to promote dietary management. Key messages from this review are that there is a need for real food intake monitoring and one of the main challenges for diet monitoring using wearable sensors is practicability when used in a different environments and how automatic dietary monitoring is important to document nutritional intake habits to prevent conditions.

Food image recognition is a difficult task due to the amount of variation within food types. Food items in images are usually accompanied with other food items as well as other unrelated non-food items. The high variation of colour, shape, size, and texture in food items means that one method of image feature extraction and classification may not adapt to other foods and therefore a feature combination approach may be needed. Conventional ways to classify images utilise the use of hand-crafted feature extraction, e.g. global or local feature extraction using Speed-Up-Robust Features (SURF) [38] or local binary patterns (LBP) [39]. Feature engineering is used to determine what type of features and parameters are best used to successfully classify certain food types and categories and much work has been completed in this area. In [5] a bag-of-features model was proposed that used a combination of scale invariant feature transform (SIFT) features along with hue-saturation-value (HSV) colour features and a linear SVM to classify images into 11 categories with 78% accuracy. Other works also utilise a combination approach using SIFT and SPIN features and achieve high accuracy in classifying high level food groups (89% accuracy in classifying sandwiches and 91.7% in classifying chicken) using Pittsburgh Fast-Food Image Dataset (PFID). However, PFID dataset is an image dataset that was developed in a controlled laboratory environment, further works could be completed in applying this feature combination approach to similar image categories photographed in real-world environments. Other works use feature selection methods to determine optimal features [8] for food image classification. As well as using traditional feature extraction methods, CNN methods have become increasingly popular for image classification and this can be attributed to ImageNet Image Large-Scale Visual Recognition Challenge (ImageNet ILSVRC) as it allows users to compete against each other in achieving a classification accuracy and the winners in recent years

ACCEPTED MANUSCRIPT

have used convolutional neural networks (CNNs). Great emphasis has been placed on using CNNs for image classification and this is evident in a surge of recent research in this area relating to the fine-tuning CNN [11], deep feature extraction [12], and also training CNNs from scratch [11].

2.1 Detecting Food in Images Using CNN

CNN has been utilised for food image detection. This problem can be condensed down to a simple binary classification problem (food/non-food). The purpose of food image detection process is to first determine if food is present within an image or video. In regards to a food image recognition pipeline, this would be the first stage in food image recognition framework i.e. determining if the image contains food or not. In [13] GoogleNet pretrained model was fine-tuned using Food-5K dataset. The training process in [13] utilised a subset of Food-5K data using 1000 iterations. The learning rate was changed to of 0.01 and the learning rate policy was polynomial. Results from [13] achieved 99.2% accuracy in determining food/non-food classes. Other research also utilised CNNs for food detection [14] and used 6-fold cross validation with different hyper-parameters to determine optimal settings and experiments achieved 93.8% in food/non-food detection.

2.2 Predicting Food Type in Images Using CNN

Extensive research has been carried out in utilising CNN for food item recognition. The food item recognition process would take place after the food detection phase in which the actual food item is then predicted within the determined food image. In [15] CNNs were utilised to extract features from convolutional layers in order to determine if an image contains a food item and experiments achieved 70.13% for 61 class dataset and 94.01% for 7 class datasets, these experiments used AlexNet deep features with a SVM classifier applied to PFID dataset [15]. In [16] the aim of the work was to compare conventional feature extraction methods with CNN extraction methods utilising UEC Food 100 dataset. Results from [16] achieved 72.6% accuracy for top-1 accuracy and 92% for top-5 accuracy. Also in [14], as well as performing food/non-food experiments, food group classification was performed. A CNN was developed and was trained using extracted segmented patches of food items [14]. The food items used in this work were based around 7 food major types. The patches were then fed into a CNN using 4 convolutional layers with different variations of filter sizes and using 5 x 5 kernels to process the patches. Results in [14] achieved 73.70% accuracy using 6-fold cross validation. These studies confirm that CNN provide an efficient method for food image recognition to provide for accurate food logging to promote dietary management.

2.3 CNN Deep Feature Extraction Methods for Food Detection/Food Item Classification

Recent research has focused have used deep features extracted from pretrained CNN architectures to train machine learning classifiers for food image classification. Some research have opted for deep feature extraction opposing to fine-tuning pretrained CNN or training from scratch because less computational power and time is needed or small image datasets are used. Well-known CNN architectures (e.g. AlexNet, VGG-16, GoogleNet) for deep feature extraction have been developed in classifying images to automate food logging. This section discusses

ACCEPTED MANUSCRIPT

research that use deep feature extraction to detect food in images and classify food items in images for automated food logging. A comparative review was carried out on analysing the performance of a number of pretrained CNN architectures [43]. This review used VGG-S, Network in Network (NIN), and AlexNet for deep feature extraction to train food detection models. A food/non-food image dataset was collated and deep features were extracted from the models to train machine learning classifiers (one-class SVM classifier and binary classifier). Results showed that binary SVM classifiers trained with deep features achieved 84.95% for AlexNet, 92.47% for VGG-S, and Network In Network model achieving 90.82%. It is worth noting that UNICT-FD889 dataset used for deep feature extraction in [43] contains minimal noise as the images are focused on the food item, therefore this may contribute to high accuracy results. Further work could be completed in utilising a larger food image dataset consisting of images from different environments and also using different machine learning classifiers for further comparison.

Other research also explored the effect of training machine learning classifiers from different layers in pretrained AlexNet architecture [15]. Authors used AlexNet model to extract deep features from various layers deep in the architecture (FC6, FC7, and FC8 layers). The food image dataset used in [15] was PFID. Two experiments were presented in [15]; classifying high-level food catergories by organising PFID dataset into 7 category dataset and also classifying individual categories in PFID (61 classes). Results showed that the highest accuracy for the 61 class dataset was 70.13% using deep features extracted from layer FC6 in AlexNet. For the 7 class dataset, the highest accuracy achieved for deep features was 94.01% using layer from FC6. The contribution in [15] echoes the same findings in [43] suggesting that deep feature extraction provides high accuracies in classifying small grouped food image datasets (related food items) as well as datasets with specific different food types. Results also suggest that AlexNet deep features are able to efficiently generalise between high level food groups and also classify specific food groups with reasonable accuracy. However, more research needs to be completed in using deep features to classify food images in real world environments as PFID used in [15] was a laboratory prepared dataset. As AlexNet is an early CNN architecture with a small amount of layers in comparison to more recent models, it was able to achieve reasonable accuracy in food item classification. AlexNet deep features from FC7 layer were able to achieve 57.87% using a standard linear SVM classifier classifying UEC-FOOD100 and 43.98% in classifying UEC-FOOD256 [45]. Fine-tuning AlexNet on a food image dataset and then performing deep feature extraction improved the accuracy to 67.57% in classifying UEC-FOOD256.

GoogleNet Inception CNN has also been used for deep feature extraction for food image classification [44]. Authors fine-tune a pretrained GoogleNet model using a food image dataset, and then deep feature extraction was used on another food image dataset. Experiments were completed in training a SVM using GoogLeNet deep features, in which the GoogLeNet model was fine-tuned using a food image dataset. Results showed that using deep features with SVM with PCA trained using fine-tuned GoogleNet features achieved 95.78% in classifying RagusaDB test set and 98.81% in classifying FCN test dataset which was an increase in accuracy comparison to other works using same datasets. Using RagusaDB and FCD combined together for experiments achieved 91.41%. The datasets used in [44] were small and more comparative research is needed in using a larger dataset of images photographed in different environments and real-world settings to fully evaluate the proposed approach [44].

ACCEPTED MANUSCRIPT

In summary, previous research has showed that deep CNN features achieve high accuracies in determining food/non-food classification and classifying high level food groups[15,43,44,45]. It is also clear from the literature that deep CNN features from various CNN architectures at varying depths can easily distinguish between food/non-food and high level food groups. It has been suggested that deep features extracted from CNN should be an initial option in any visual recognition tasks [51], however in regards to food image classification, more work needs completed in exploring the use of next generation CNN architectures to extract deep features to train food classifiers, primarily for specific food item image classification photographed in real-world environments. This work compared the performance of using ResNet-152 and GoogleNet CNN deep features to classify a variety of food image datasets for food logging applications.

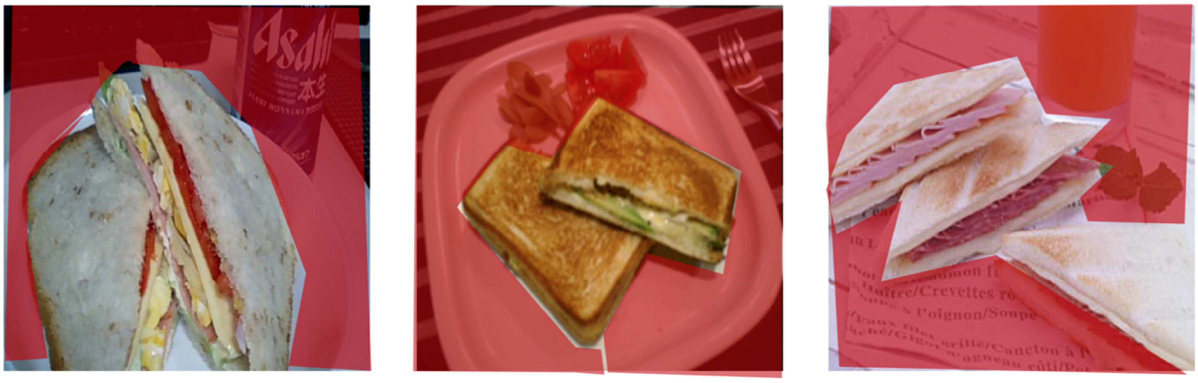


Figure 1: Example images of sandwiches from UEC FOOD 256 dataset highlighting noise in images.

3. Aim & Objectives

The aim of this work was to investigate the effectiveness of using deep feature extraction methods to classify variety of food image datasets to be used for dietary assessment. The work described in this paper seeks to answer the following research questions:

1. How efficient are deep residual network features for detecting foods in images and classifying food datasets using conventional machine learning algorithms?

2. How efficient are extracted GoogleNet deep features in predicting Food/Non-Food images and classifying images into high level food groups in comparison to fine-tuned GoogleNet model?

A series of experiments were completed that used the features extracted from CNNs and used them as input into conventional machine learning algorithms. To answer the research questions a number of objectives needed to be completed to achieve the aim of this work: (a) a number of public food image datasets needed to be selected, (b) several pre-trained CNNs needed to be identified from the literature for deep feature extraction, (c) supervised machine learning algorithms needed to be identified to classify the images using the extracted deep activations; and (d) statistical analysis is then applied to the results to evaluate the methods used. The next section will discuss in detail the methods used in this work.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4. Methodology

4.1 Food Image Datasets

In this work we identified publicly available food image datasets to use for the experiments to determine efficiency of using pretrained CNNs to extract deep features for image classification. The following image datasets were used in this work (Table 1):

1. Food-5K
2. Food-11
3. RawFooT-DB
4. Food-101
5. UNICT-FD889
6. Caltech-101

Table 1: Table showing name, number of categories, images per category, as well as how the image datasets were developed of each food image dataset.

Dataset	Categories	Images per Category	Image Preparation
Food-5K	2	2500 (training set) 500 (val & eval sets)	Real world
Food-11	11	Unbalanced	Real world
RawFooT-DB	68	368 each in training/testing set	Controlled/ Laboratory
Food-101	101	1000	Real world
UNICT-FD889	889	Unbalanced	Real world
Caltech-101	101	Unbalanced (non-food)	Real world/Controlled

ACCEPTED MANUSCRIPT

4.2 Food-5K

Food-5K dataset consisted of 2 categories; food and non-food, training is balanced and contains 2500 images of each category [13]. The dataset also contains a validation and evaluation set and each category contains 500 images each per dataset. The authors developed this dataset to measure the performance of using a fine-tuned GoogleNet pretained CNN for classification. Food-5K was developed by selecting images from already public available datasets e.g. Food-101 [17], UEC-FOOD100 [18] and UEC-FOOD256 [19]. The authors described this dataset as being varied as they selected foods that cover a wide variety of different food dishes. The images also contain some noise and multiple food items may be contained in an image. The non-food images consisted of images that do not contain food items (objects or humans). Food-5K was used to find out how ResNet-152 deep features perform in detecting food items in images, which can be argued is an important first step in food image classification for food logging. The authors developed the non-food image dataset from using other publicly available datasets e.g. Caltech101, Caltech256, Emotion6, and Images of Groups of People.

4.3 Food-11

Food-11 is a dataset that comprises of 11 major food groups [13]. The 11 categories are diary, bread, egg, dessert, meat, fried food, pasta, seafood, rice, vegetables/fruit, and soup. Food-11 dataset was also created using images from Food-101, UEC-FOOD-100, and UEC-FOOD-256. The authors of Food-11 stated that the images selected cover a wide range of food types in order to train a strong classifier that had the ability to classify different varieties of foods. Many of the images contained in Food-11 were taken in real world environments, therefore the images contain high colour variation and some noise (unrelated food items) may be present. The developers of this dataset have divided the dataset into training, validation, and evaluation similar to Food-5K. Food-11 was used to explore the performance of ResNet-152 deep features in categorising food images using Food-11.

4.4 RawFooT-DB

RawFooT-DB [20,42] food image dataset was developed to research the use of computer vision methods to classify food image textures under different lighting conditions. Each image in RawFooT-DB is unique in regards to the light direction, light intensity, and colour illumination and food image textures are isolated with no noise or other food items present. The dataset contains 68 classes with wide variety of food types ranging from fish, meat, fruit, and cereals. RawFooT-DB dataset contains tiles from the images in the RawFooT-DB. Each image is divided into 16 tiles, 8 tiles are for training and the remaining 8 for testing. Each class contains 368 images (tiles) which represent 8 tile texture samples under 46 different lighting conditions. In this research, we explored the use of ResNet-152 deep feature features to train machine learning classifiers. RawFooT-DB was used to explore how ResNet-152 deep features perform in generalising food texture between class variance. Previous research divided RawFooT-DB into different lighting condition subsets [20, 42], in this work we explored the performance of using

ACCEPTED MANUSCRIPT

ResNet-152 deep features across multiple lighting conditions and each food class in RawFooT-DB contains multiple food texture patches across different lighting conditions.

4.5 Food-101

Food-101 consists of 101 food categories and each category contains 1000 images [17]. The Food-101 dataset have been described as challenging as much of the images in the dataset contain noise and the images were collated from Foodspotting, which is a social media website that allows users to upload food images. This means that images used are from a real-world setting i.e. restaurant or at home and not in a lab environment. Food-101 allows us to research how ResNet-152 deep features perform in classifying food items with similar food dishes in varying real world environments. Authors of Food-101 specify dedicated training and testing splits with testing splits containing images that are 'cleaned' of noise, in this work we also use 75:25 training/testing partitions, however data was shuffled before partition for preliminary analysis to determine how ResNet-152 features perform in classifying images with noise and intense colour and food variation. Figure 2 illustrates an example of the images in the datasets.

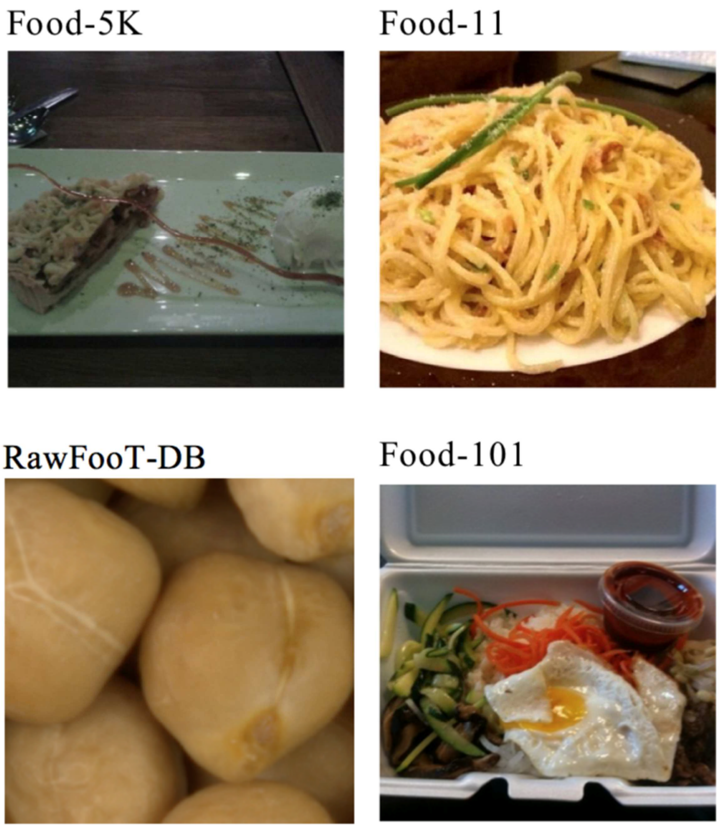


Figure 2. Image examples from 4 food image datasets used in this work.

4.6 Datasets for Further Evaluation of Food/Non-Food Detection Models

ACCEPTED MANUSCRIPT

Due to the small size of Food-5K, two other datasets have been used to evaluate our trained food/non-food models; UNICT-FD889, which is a food image dataset, and Caltech-101, which is a non-food image dataset. Deep features were extracted from UNICT-FD889 and Caltech and classified by models that achieved the best performance in classifying Food-5K datasets.

UNICT-FD889

UNICT-FD889 (Figure 3) was used to evaluate food/non-food models trained using Food-5K [53]. UNICT-FD889 contains 889 distinct food dishes to study food representation and the images are photographed in real world environments which means that much of the images may contain high food variance, however the images in UNICT-FD889 contain images that are focused on the food item with little noise

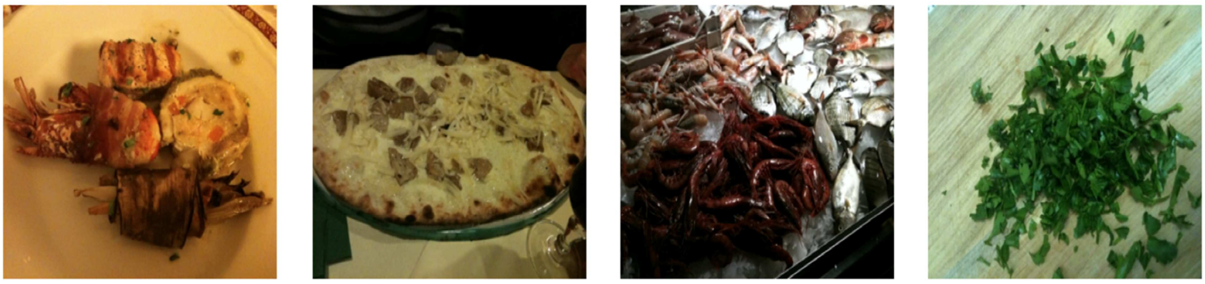


Figure 3: Example of images contained in UNICT-FD889 dataset.

Caltech-101

Caltech-101 dataset (Figure 4) was also used for evaluating food/non-food classification models. Caltech-101 contains 101 image categories and each contains between 50-800 images. The categories are non-food based and contain images relating to animals and objects and each image is around 300x200 pixels in size [52].

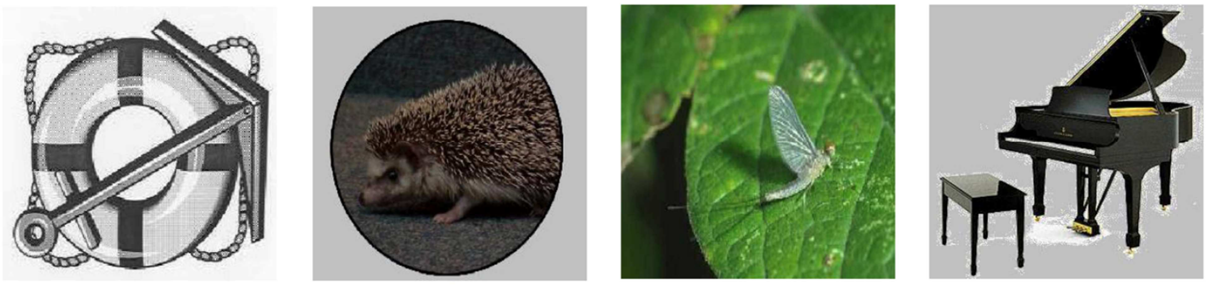


Figure 4: Example of images contained in Caltech-101 dataset.

4.7 Overview of Convolutional Neural Networks

The use of pretrained CNNs gives great potential for applying them to a variety of problem areas. Convolution is used to describe the type of neural network as the input image is broken down into smaller overlapping shapes in order to determine certain patterns in the image. These overlapping segments are called filters. The patterns detected, by each overlapping shape in the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

filter, may consist of a colour contrast or certain interest points such as edges. The overlapping shapes look for the same pattern on the image. The overlapping tiles are effectively used as input for a small neural network. This is done for each tile in the image. Each network in the filter hold the same weights to determine interest points in each tile. The output of this process is an array which each section corresponds to the network that describes patterns in each tile. A down-sampling process is then triggered after the convolution stage, this is typically completed using max pooling where the representation divided into non-overlapping rectangles. Within each region the maximum is retained. This process can be repeated a number of times to create deeper and more detailed representations. Fully connected layers are also present with a CNN architecture and is connected to activations from the layer previous. The fully connected layer takes the input from previous layers and uses this for classification using a soft-max function. Backpropagation is typically used to train the CNN in which the forward propagation is used to determine the error and gradient descent is then used to update the weights and parameters based on this error. This is repeated in order to train the CNN using a training dataset [21,22].

4.8 Image Preprocessing for Feature Extraction

The pretrained CNNs used in this work were trained specifically with requirements placed on the input images. Therefore, in order to extract deep feature representations of these images using these CNNs, it was important to ensure that the images meet the same requirements. The first requirement was to ensure that the images were resized to a specific height and width configured in the image input layer of the pretrained CNN. The images are also normalised and this is achieved by subtracting the mean of the image. The mean is removed from the input image and also the image intensities are normalised within a [0,255] region, as defined in [23].

4.9 Deep Feature Extraction

In this work we used 2 pretrained CNNs as deep feature extractors. The ad- vantage of using a pretrained CNN to extract deep image features, as opposed to training a new CNN, are: (1) less computational power is needed as we are allowing the CNN to process each image only once to extract deep feature representations; (2) less data is needed in order to achieve high accuracy results as layers deep in the CNN architecture contain activations that can be used for deep feature representations.

CNNs have been trained to specifically determine and highlight key features in an image and pretrained CNNs allow images to be inserted and layers produce a response or activation to the image. These ’activations’ or deep features as they will be called in this work, can be extracted in the form of a feature vector [23,24]. The authors that created datasets Food-5K and Food-11 fine- tuned a GoogLeNet model, therefore for performance comparison, we adopted a different approach of using GoogLeNet, not for fine-tuning but for deep feature extraction and to use these deep features to train machine learning classifiers. As stated, the 2 CNNs we have chosen achieved high accuracy results when applied to ILSRVC ImageNet dataset.

Comparing this feature extraction process to training a CNN from scratch, in which mini-batches of image data are iteratively passed through different layers (i.e. convolutional and sub-sampling layers) using back-propagation to implement stochastic gradient descent to train the network, the method of deep feature extraction requires less computational power. Deep feature extraction can also be implemented on a CPU as only one pass is completed through the training data to extract the deep features. It is also worth noting that a large amount of time needs to be dedicated to train a CNN from scratch. For many researchers this is not possible, therefore pretrained CNNs offer a convenient way to experiment with deep learning algorithms by allowing for deep feature extraction, classification, and also transfer learning.

The datasets used in this work are small in comparison to the datasets needed to train a CNN from scratch such as ILSRVC dataset which contains over 14 million images [59]. Figure 5 describes the pipeline used in this work where by images are processed to extract deep features to be used for classification.

4.9.1 Layer Selection

To extract features from pretrained CNN, a layer needs to be selected for each model. During the training of CNN models, the output from convolutional layers and the pooling layers depict high level representations of images. In this study we extracted deep feature maps immediately after the last pooling layer of each CNN to determine if these feature representations are able to accurately generalise between different food classes in food image dataset. The layer names used to extract deep features from CNN architecture are used to distinguish between different layers in the pretrained CNN models. Table 2 lists the size of each pretrained CNN model and the chosen layer for deep feature extraction.

Table 2: Table showing pretrained CNN used as deep feature extractors in this work. The table lists the name of the CNN, the amount of layers present, the dataset used to train the CNN, and layer used in this work.

CNN	**Layers**	**Trained Using**	**Layer**
ResNet-152	152	ImageNet ILSVRC	pool5
GoogLeNet	22	ImageNet ILSVRC	cls3_pool

4.10 Pretrained Models using MatConvNet Package

MatConvNet is a popular Matlab library that allows for the training of state- of-the-art CNNs or to apply pretrained CNNs for deep feature extraction to be used for image classification [23,24]. In this work, MatConvNet was used to utilise 2 pretrained CNNs for deep feature extraction both trained on ILSVRC ImageNet dataset. MatConvNet packages allow for the fine-tuning of pretrained CNN [24]. In this work ResNet-152 and GoogLeNet were chosen to extract deep features to train classification models, the reason ResNet-152 was used was that it has achieved the lowest top-1 error of 23% using ILSVRC 2012 validation dataset in the MatConvNet package. GoogLeNet is another popular model available on MatConvNet package and was used

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

for deep feature extraction in this work for performance comparison with the fine-tuned GoogleNet model trained in [13].

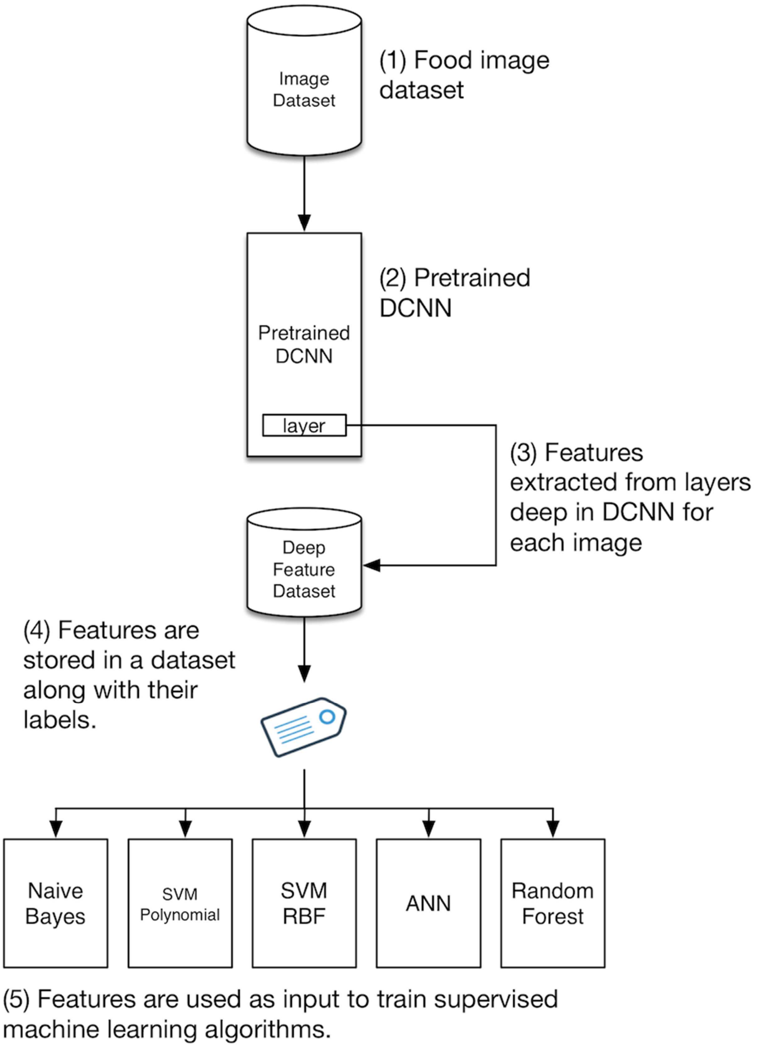


Image Dataset
(1) Food image dataset
Pretrained DCNN
(2) Pretrained DCNN
layer
(3) Features extracted from layers deep in DCNN for each image
Deep Feature Dataset
(4) Features are stored in a dataset along with their labels.
Naive Bayes
SVM Polynomial
SVM RBF
ANN
Random Forest
(5) Features are used as input to train supervised machine learning algorithms.

Figure 5: Diagram describing the pipeline of deep feature ex- traction. (1) Food image datasets are used as input into (2) (pretrained CNN). (3)A layer deep in the architecture is specified and the image is processed by the CNN and the output (of the specified layer) is a generic image feature vector. (4) These generic image feature vectors can be collated to form a feature dataset and each feature vector generated by the CNN layer is labelled in accordance to the category from where the image taken from. (5) The generic image feature dataset can then be used as input to a range of conventional machine learning algorithm.

4.11 ResNet-152 CNN

ACCEPTED MANUSCRIPT

ResNet-152 is a deep residual pretrained CNN [25]. At the time of develop- ment, the authors of this CNN have described it as the deepest network ever presented on ImageNet (2015) and is based on utilising extremely deep nets with a depth of up to 152 layers. A residual learning framework which allows training of networks easier to converge and promote increased accuracy. The main advantages that residual networks contribute is the acceleration of speed in training networks, the effect of the vanishing gradient problem is reduced, and increasing the depth of the network which results in less parameters. ResNet- 152 is made up of residual connections that allow important information to be transferred between layers. Residual connections allow a gradient to pass backwards directly through layers without losing vital information, in a regular CNN, the gradient must always pass through an activation layer. This can cause the gradient to diminish, to circumvent this problem, connections within a CNN are appended with a shortcut that allows gradients to pass through thus decreasing the effects of vanishing gradient (information loss). Experiments us- ing residual connects (ResNet-152) have reported increased accuracy and lower training times, in comparison to other state of the arts [25]. The authors of ResNet-152 compare their work with other established CNNs and state that this residual deep net is 8x deeper than VGG nets [26]. We used ResNet-152 pretrained CNN with the image datasets mentioned in this work for feature extraction. We selected pool5 layer deep in the ResNet-152 architecture and for each image an extracted a feature vector of 2048 was computed.

4.12. GoogleNet - Inception

GoogLeNet was used for deep feature extraction combined with the same supervised machine learning models. In [22] a deep convolutional network was proposed that is able to achieve state of the art classification and object detec- tion accuracy by training the network using ImageNet dataset for Large Scale Visual Recognition Challenge 2014. The motivation for GoogLeNet was that larger CNNs may encounter the problem of overfitting as there is a large number of parameters used in the network. GoogLeNets main contribution is the intro- duction of Inception modules that utilises the concept of using approximation of sparse structure with repeated dense components. Dimensionality reduction is used in order to ensure computational complexity is kept to a minimum. Mul- tiple convolutional filters are used with different sizes to ensure that there is sufficient coverage of information clusters. Before more computational expen- sive convolutions (3x3, 5x5) a convolutional after the previous layer for data reduction. The results of GoogLeNet incorporating these inception modules achieved 6.67% top-5 error percentage in classification performance in ILSVRC Classification Challenge 2014. In this work, we extracted the deep activations using the fully connected layer cls3 pool which has a 1024 vector dimension and is located after the last pooling layer in GoogLeNet [22].

4.13 Metrics for Performance Measurement

Several metrics were used to assess the performance of the trained models. The metrics that were selected to assess each model were percentage, recall, F1 score, Kappa, and Area Under the Receiver Operating Characteristic curve (AUC). The output of each model can be presented using a confusion matrix. A confusion matrix is a table that is able to summarise the prediction outcome of a model by classifying instances as positive (P) instances or negative (N) instances. Confusion matrix can further provide greater insight into prediction outcomes by classifying predicted instances as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Visually, the performance of a confusion matrix can be quickly assessed by inspecting the diagonal line of the confusion matrix, the stronger instances that are present in this diagonal line signifies better performance. The metrics used to assess the experiments can be derived from the confusion matrix such as recall (sensitivity), Ac, and F1 score. Recall can be described as metric that describes how many instances are classified correctly. The F1 score is a weighted average using precision and recall and is measured between 0 (worst) and 1 (best). For Food-5K the AUC values were also computed for each experiment due to being a binary classifier and Cohen's kappa was calculated for Food-11, RawFooT-DB, and Food-101. Cohen's kappa is a metric that is used to measure the inter-rater agreement between two label sets in a classification problem, we use Cohen's Kappa along with other metrics to describe experiment results [27].

4.14 Training, Validation, and Evaluation Data Partitions

To evaluate the performance of our trained models, validation and evaluation datasets were extracted and used from Food-5K, and Food-11. For RawFooT- DB, an evaluation dataset was used supplied by the authors [20]. For Food-5K, Food-11, and RawFooT-DB, the authors already partitioned the datasets into evaluation and validation sets (Table 3) and in this work we used the same data splits to train and test our models. For Food-101, we split the data into 75:25 for training and testing. Authors of Food-101 provide training and testing splits with testing images cleaned of noise, however in this work we randomly shuffled images for training and testing partitions to test how ResNet-152 performs in classifying food images with noise and high food variance. This would give an indication of how deep features would perform in classifying difficult datasets such as Food-101. Table 3 is a summary of the data partitions used in this work for each food image dataset and the names for each partition follows the author's naming convention. Several metrics were computed during the experiment stage e.g. kappa statistic, F1 score, recall, average ROC, and accuracy to measure the performance of each trained model. Food-5K and Food-11 datasets each contained training, validation, and evaluation images. Training images were used for feature extraction to train machine learning classifiers. Validation images were used to determine if hyper-parameters used yield adequate results and evaluation dataset was to fully evaluate overall trained model. For RawFooT- DB, authors developed training and testing datasets by taking each image and dividing it into 16 tiles, 8 tiles are for training and the remaining 8 for testing. Each class contains 368 images (tiles) which represent 8 tile texture samples under 46 different lighting conditions. The testing dataset was used to verify if the trained model able to generalise between food texture classes. Food-101 dataset was randomly partitioned; 75% for training and 25% for testing. Testing partition was used to verify trained Food-101 classifiers. UNICT-FD889 and Caltech-101 testing datasets were used to further evaluate food/non-food classification models.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 3: Table showing testing methods used for each food image dataset. * denotes dataset splits supplied by dataset authors.

Dataset	Dataset Partition
Food-5K	Training, validation, & evaluation*
Food-11	Training, validation, & evaluation*
RawFooT-DB	Training & testing*
Food-101	75:25 training & testing
UNICT-FD889	Testing
Caltech-101	Testing

4.15 UNICT-FD889 & Caltech-101 Food/ Non-Food Dataset

As well as using the validation and evaluation datasets supplied with Food- 5K, further evaluation was completed with UNICT-FD889 dataset and Caltech- 101 dataset in detecting food images. UNICT-FD889 is a food dataset containing images from a range of food types and Caltech-101 is a non-food image dataset, UNICT-Caltech. These 2 datasets were combined to create a new food/non-food dataset called UNICT-FD889 to evaluate our food detection models. Deep features were extracted from the new food/non-food dataset. Further evaluation was completed because Food-5K evaluation and validation datasets are small with only 500 images in each category for each dataset. Using another larger dataset for evaluation can give a stronger performance indication of our models in classifying a large variety of food and non-food images.

4.16 Weka Platform

In order to train the machine learning algorithms, Weka 3.8.1 [28] platform was used. Weka is a software application that contains various machine learning algorithms written in Java and the application was developed at University of Waikato, New Zealand. The application can be used for different tasks such as clustering, classification, visualisation, feature selection, and preprocessing and is very popular within universities for its ease of use. It is also popular because of the amount of algorithms available. The main reason that Weka 3.8.1 was used in this

ACCEPTED MANUSCRIPT

work was the detailed evaluation results output computed, which are collated into a window after evaluation has finished. Another major advantage of using Weka is the evaluation process in that a range of detailed metrics are computed for each class to describe the performance of the model. A confusion matrix can be computed to determine the performance of individual classes for the trained model using K-fold class validation or a dedicated validation dataset. The amount of machine learning algorithms that are available is a factor in using Weka as well the easy to use graphical user interface (GUI). In this work, Weka 3.8.1 was used with the extracted features from image datasets for classification, analysis, and evaluation [28].

4.16.1 WekaPython Plugin & Scikit-Learn

WekaPython plugin was used with Weka 3.8.1 that allows the training of scikit-learn [29,55] machine learning classifiers. The wekaPython package relies on Python version 2.7 or higher being installed on the user's system and uses a range of Python packages to function correctly such as pandas, numpy, scikit- learn, and matplotlib. In this work, the wekaPython was used to train and evaluate the deep features extracted from the pretrained CNNs. Weka was used to train an ANN for experiments with Food-101. Due to its flexibility for working with larger datasets, Python v2.7.10 with scikit-learn library was also used to train the other machine learning classifiers for the Food-101 dataset [30]. The following machine learning algorithms were used in this work [29,54]:

1. Gaussian Naive Bayes (wekaPython scikit-learn)
2. Support Vector Machines (SVM) (wekaPython scikit-learn)
3. Artificial Neural Network (ANN)
4. Random Forest Classifier (wekaPython scikit-learn)

For Food-101 food image dataset, datasets were manually split 75:25 and the follow parameters were used to split and shuffle the dataset to train and test each machine learning classifier;

1. Gaussian Naive Bayes - random state 1
2. Support Vector Machines - random state 1
3. Artificial Neural Network - random seed 1
4. Random Forest Classifier - random state 1

4.16.2 Naive Bayes

Naive Bayes is a popular machine learning algorithms known for their efficiency and minimal processing. They can be described as a set of simple probabilistic classifiers derived from Bayes Theorem. The term naive is used to describe the algorithm because it assumes that attributes are independent of the associated class. Bayes rule is enforced to compute the probability of a class based upon the values in the vector. Bayes rule of conditional probability states that if you have a hypothesis H and the evidence (feature attributes) is connected to that hypothesis [31]. Naive Bayes assumes independence and the algorithm works efficiently and can outperform the most sophisticated machine learning algorithms on certain datasets. Naive Bayes can be described as a simplistic approach to using learning probabilistic knowledge for classification. However, the present of redundant data can affect the performance and the introduction dependent attributes

ACCEPTED MANUSCRIPT

also diminish the performance of classifier. In this work, a Gaussian naive bayes classifier was trained using the extracted CNN deep features. A Gaussian naive bayes classifier is used when continuous values are present by assuming a normal distribution in the dataset as the mean and standard deviation is computed for each class.

4.16.3. Support Vector Machines (SVM)

SVMs are able to implement the use of non-linear boundaries by using ker- nels (e.g. RBF, Polynomial) to transform feature representation into a higher dimensional space to predict multiple classes. In classification problems, the use of SVM have performed well in generalising on a variety of classification problems such as food classification, face detection, and object detection [32,33]. In some problems the training data in a problem may become inseparable meaning that there is not a clear boundary definition, SVMs are able to enforce nonlinear boundaries in transformed feature spaces [35]. In regards to a linear SVM, a linear hyperplane is computed and considered optimal if a line is at a furthest distance from class data points (largest minimum distance) [35]. However, in some instances the training data may not be linearly separable, therefore SVM employ the use of kernels to determine optimal hyperplanes. Kernels can be used in order to fit linear models in a non-linear setting, mapping is used to transform how the features are represented into a higher dimensional space. In this work, we train 2 C-SVM models using Polynomial kernel and Radial Basis Function (RBF). C-SVM uses a C regularisation parameter that implements a weight penalty for misclassifications to improve the accuracy of the model.

4.16.4. Artificial Neural Network (ANN)

An ANN or feed-forward neural network was also used in this work and ANN can comprise of a number of layers. Each layer contains a number of nodes that are called neurons. The basic ANN architecture is made of three layers; input layer, hidden layer, and output layer and because of the amount of rich information/features that can be learned using a ANN, it can be applied to problems that are of an non-linear nature. The basic function of a ANN is the ability to map features data into a set of outputs. Each neuron computes its input by using a weight that represents the strength between nodes. An activation function is then applied, there are a number of activation functions that are available i.e. sigmoid function, linear, or Gaussian. Once the activation function is applied, a single value is returned. Back propagation is used to train the ANN, the predicted output is compared to the expected output which is reflected in the cost function and the weights are altered. ANN training can be customised to suit the nature of the input dataset and problem, parameters such as training time (epochs), learning rate, and momentum can be configured. In this work, ANNs were trained for each dataset using a Weka plug-in [30] with the following parameters listed in Table 4. The learning rate was set to adaptive unless otherwise stated in the experiments. The adaptive learning rate function uses a number of base learning rates on the training data to determine the most suitable by comparing the cost function of each. The Weka plugin uses dropout regularisation to prevent overfitting and Rectified Linear Units as the activation functions [30, 36].

Table 4: Hyper-parameters used for each ANN.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

ANN	Parameters
Number of iterations	1000 (max)
Number of layers	1
Neurons per layer	100
Learning rate	Adaptive*
Learning momentum	0.2
Weight Penalty	0.00000001 (default)
Hidden Layers drop out rate	0.5
Input layer drop out rate	0.2
Activation function	ReLu
Convergence threshold	0.2
Batch	100

4.16.5 Random Forest

Random Forests (RF) was developed by Leo Brieman and Adele Culter [37] and is a classification algorithm that utilises a number of decision trees using feature subsamples and bootstrapped examples. The purpose of RF was to be easy to use by offering little preprocessing requirements and using a voting system for final classification using a collection of decision trees. This method is directly related to the bagging technique as the goal of the bagging technique is to develop a model with low variance and to average noise in the dataset. RF is able

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

to take subsets of the input data comprised of random values with each instance labelled with its class. For each subset created a decision tree is created. Each decision tree is trained using the subset training data and a classification for each instance is calculated. A majority voting rule is then used to decide on the final classification of the instance. RF algorithm is efficient in that it is able to analyse large databases and is able to estimate missing data to help maintain accuracy [37]. In this work a scikit-learn RF classifier was used with wekaPython and Table 5 lists the parameters used for this model.

Table 5: Table showing hyper-parameters used for WekaPython Random Forest classifier. Hyper-parameters used for this classifier are default.

Random Forest	**Parameters**
Criterion	Entropy
Number of estimators	50
Random state	None
Depth of tree	None
Minimum number of samples split	2
Minimum number of samples for leaf node	1
Number of features for best split	auto
Bootstrap	True
Max leaf nodes	None
Random state instance	None
	None

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Max depth	
Minimum num of leaf samples	1

5. Experimental Results

5.1. Food /Non-Food Classification Results

5.1.1. Food-5K

This section lists the results of our experiments using the food image datasets. Tables 6 and 8 list the detailed results of Food-5K. Accuracy, recall, F1 score, and ROC values were used to measure the performance of each the classification models for both validation and evaluation datasets. Initial results show that deep features combined with machine learning classifiers achieved high accuracy results when distinguishing between food and non-food images. The use of SVM with RBF kernel achieved the highest accuracy with 99.4% using ResNet-152 for deep feature extraction with validation dataset and 98.8% with evaluation dataset. Table 7 and 9 also lists the confusion matrices of using SVM-RBF with ResNet-152 to detect food images in validation dataset and ANN with ResNet-152 features to detect food images in evaluation dataset. GoogLeNet deep features achieved marginally lower accuracy results, however for the evalu- ation dataset, GoogLeNet deep features with ANN achieved the same accuracy result as SVM-RBF and Random Forests classifier with ResNet-152 features with 98.8%. In regards to using SVM classifiers in Food-5K, the use of the RBF kernel achieved marginally higher accuracies compared to the polynomial kernel and Gaussian Naive Bayes achieving the lowest accuracy results in both testing datasets with both deep feature types.

ACCEPTED MANUSCRIPT

Table 6: Classification results using ResNet-152 and GoogleNet to extract deep activations (extracted from Food-5K) with supervised learning algorithms. Figures in bold represent highest accuracy result.

Food-5K - Validation								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3 pool**			
	Acc (%)	**Recall**	**F1**	**ROC**	**Acc (%)**	**Recall**	**F1**	**ROC**
NB	98.7	0.99	0.99	0.99	97.5	0.98	0.98	0.99
SVM (RBF)	99.4	0.99	0.99	0.99	98.5	0.99	0.99	0.99
SVM (Poly)	99	0.99	0.99	0.99	98.5	0.99	0.99	0.99
ANN	99.2	0.99	0.99	1	99	0.99	0.99	0.99
RF	98.9	0.99	0.99	1	98.6	0.99	0.99	0.99

Table 7: Confusion matrix showing results of highest accuracy results achieved using ResNet- 152 features classifying **validation** dataset of Food-5K using a SVM with RBF kernel.

		Predicted Labels	
		Food	Non-Food
True Labels	Food	**498**	2
	Non-Food	4	**496**

Table 8: Classification results using ResNet-152 and GoogLeNet to extract deep activations (extracted from Food-5K) with supervised learning classifiers using evaluation dataset.

ACCEPTED MANUSCRIPT

Food-5K - Evaluation								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3 pool**			
	Acc (%)	**Recall**	**F1**	**ROC**	**Acc (%)**	**Recall**	**F1**	**ROC**
NB	97.3	0.97	0.97	0.98	96	0.96	0.96	0.98
SVM (RBF)	98.8	0.99	0.99	0.99	98.3	0.98	0.98	0.98
SVM (Poly)	98.3	0.98	0.98	0.98	98.2	0.98	0.98	0.99
ANN	98.8	0.99	0.99	0.99	98.8	0.99	0.99	0.99
RF	98.8	0.99	0.99	0.99	98.5	0.99	0.99	0.99

Table 9: Confusion matrix showing results of highest accuracy results achieved using ResNet- 152 features classifying **evaluation** dataset of Food-5K using ANN.

		Predicted Labels		
		Food	Non-Food	
True	Food	**493**	7	Labels
	Non-Food	5	**495**	

To further test our models, experiments were conducted that tested food/non- food trained models on the Food-11 dataset as what was completed in [13] for more detailed comparison. Food-11 dataset contains 16,643 images and they are all classed as food images, GoogleNet and ResNet-152 deep features were used to extract deep features from Food-11 and used with SVM-RBF and ANN models to classify them to detect food in the images. Table 10 is a breakdown of the results using our methods to classify Food-11 dataset.

ACCEPTED MANUSCRIPT

Table 10: Results comparison of classifying Food-11 and UNICT-Caltech with our Food/Non-Food classification models.

Method	Number of food images detected	Accuracy
ResNet-152 + ANN (Food-11)	16,208	97.39%
ResNet-152 + SVM-RBF (Food-11)	16,176	97.19%
GoogleNet + ANN (Food-11)	16,171	97.16%
GoogleNet + SVM-RBF (Food-11)	15,646	94.01%
ResNet-152 + SVM-RBF (UNICT-Caltech)	12,409	97.50%
ResNet-152 + ANN (UNICT-Caltech)	12,283	96.51%

5.1.2. UNICT-FD889 & Caltech

Table 10 list the results of using SVM-RBF and ANN trained with Food- 5K training ResNet-152 deep features for classifying UNICT-Caltech, which combines images in UNICT-FD889 and Caltech-101 to make a food/non-food dataset. UNICT-Caltech dataset is a larger dataset and using this dataset with our trained models allows us to get a better indication how ResNet-152 features perform in detecting food in images.

5.2. Food Item Classification Results

5.2.1. Food-11

Results show that using ResNet-152 and GoogleNet deep features are able to achieve high accuracies when classifying across major food groups. Results are presented in Tables 11 and 12. The maximum accuracy achieved was using ANN for both ResNet-152 and GoogleNet features achieving 91.34% and 86.44% respectively with evaluation dataset. For ResNet-152 features an F-measure of 0.91 was achieved and 0.86 with GoogleNet features using ANN. For the ANN trained using ResNet-152 features, the base learning rate was set to auto-detect which allows the

ACCEPTED MANUSCRIPT

ANN Weka plugin to initially test various learning rates to determine the lowest cost function. Initial tests revealed that 1.0 learning rate achieved the lowest cost function and the ANN used that to learning rate to initially begin the training. The learning rate decreased over the course of the training if the network cost function didn't improve after 10 mini-batch iterations. The network converged after 204 iterations ending with a learning rate of 0.01. Further analysis revealed the SVM models trained with RBF and Polynomial kernel using ResNet-152 features achieved 89.99% and 88.86% accuracy respectively and 85.36% and 86.05% using GoogleNet features using evaluation dataset. Figure 6 shows the confusion matrix of using an ANN trained with ResNet-152 features to classify the evaluation dataset. Figure 7 is an example of different types of food categories that were misclassified as shown in the confusion matrix in Figure 6.

Table 11: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from Food-11) with supervised learning classifiers.

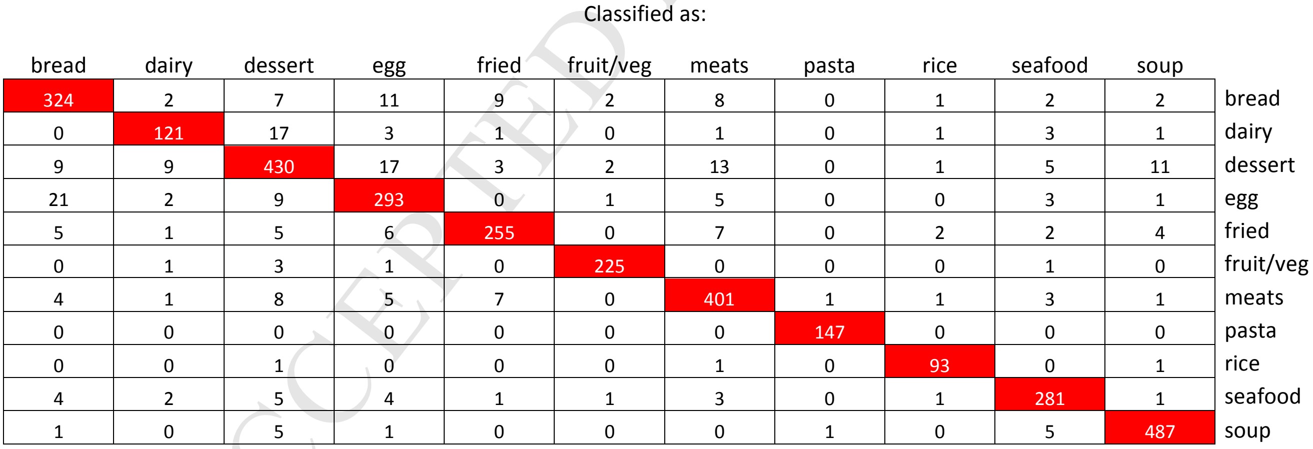
Food-11 - Validation Dataset								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3 pool**			
	Acc (%)	Recall	F1	Kappa	Acc (%)	Recall	F1	Kappa
GNB	73.03	0.73	0.73	0.70	67.49	0.68	0.68	0.64
SVM (RBF)	88.11	0.88	0.88	0.87	82.36	0.82	0.82	0.80
SVM (Poly)	86.65	0.87	0.87	0.85	83.70	0.84	0.84	0.82
ANN	89.18	0.89	0.89	0.88	84.11	0.84	0.84	0.82
RF	78.43	0.78	0.78	0.76	75.48	0.76	0.75	0.72

ACCEPTED MANUSCRIPT

Table 12: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from

Food-11) with supervised learning algorithms.

Food-11 - Evaluation Dataset								
Model	ResNet-152 - pool5				GoogLeNet - cls3 pool			
	Acc (%)	Recall	F1	Kappa	Acc (%)	Recall	F1	Kappa
GNB	75.38	0.75	0.76	0.72	69.73	0.70	0.70	0.66
SVM (RBF)	89.99	0.90	0.90	0.89	85.36	0.85	0.85	0.84
SVM (Poly)	88.86	0.89	0.89	0.87	86.05	0.86	0.86	0.84
ANN	91.34	0.91	0.91	0.90	86.44	0.86	0.86	0.85
RF	80.40	0.80	0.80	0.78	78.24	0.78	0.78	0.75



Classified as:

bread	dairy	dessert	egg	fried	fruit/veg	meats	pasta	rice	seafood	soup	
324	2	7	11	9	2	8	0	1	2	2	bread
0	121	17	3	1	0	1	0	1	3	1	dairy
9	9	430	17	3	2	13	0	1	5	11	dessert
21	2	9	293	0	1	5	0	0	3	1	egg
5	1	5	6	255	0	7	0	2	2	4	fried
0	1	3	1	0	225	0	0	0	1	0	fruit/veg
4	1	8	5	7	0	401	1	1	3	1	meats
0	0	0	0	0	0	0	147	0	0	0	pasta
0	0	1	0	0	0	1	0	93	0	1	rice
4	2	5	4	1	1	3	0	1	281	1	seafood
1	0	5	1	0	0	0	1	0	5	487	soup

Figure 6: Confusion matrix of Food-11 classes using ANN model trained using ResNet-152 features.

ACCEPTED MANUSCRIPT

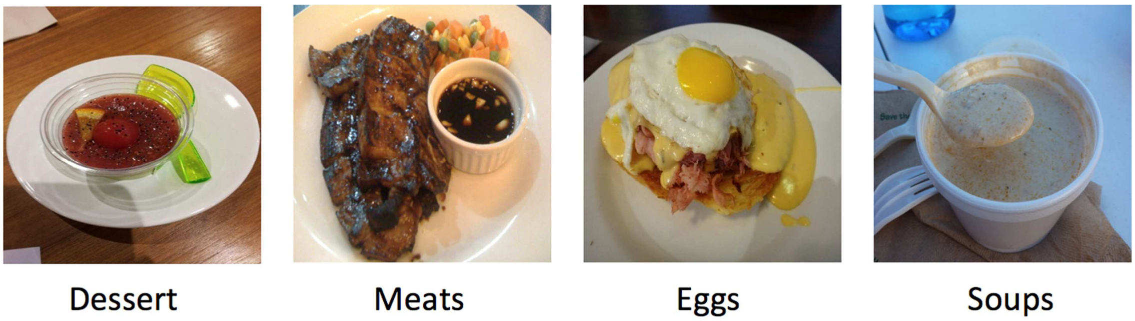


Figure 7: Example of Food-11 classes which are misclassified based on confusion matrix generated from ANN model trained using ResNet-152 features. Images highlight shared characteristics that could lead to misclassifications.

5.2.2. RawFooT-DB Classification Results

Results listed in Table 13 reveal ResNet-152 features trained with SVM and RBF kernel achieved an accuracy of 99.10% and our ANN also with ResNet- 152 99.28% in classifying RawFooT-DB. The results show that deep features efficiently classify isolated texture images across various lighting conditions and further investigation analysing the confusion matrix generated from SVM-RBF model shows that there were several classes that experienced misclassifications. For example, several instances were wrongly classified as chickpeas instead of white peas. Investigating the images from both categories, it was clear that there are similarities between shape, colour, and texture as shown in Figure 8 and 9. When also investigating the ANN confusion matrix, several white pea instances were also classed as chickpeas and there were also several mango instances classed as apple slice. Figure 9 is an example of image classes that were misclassified using an ANN, chicken breast and milk chocolate. These images showed similar characteristics in colour and texture, similarly hamburger images were classified as salami and further investigation showed very similar texture, colour, and patterns however ResNet-152 features still achieved 0.98 F-measure for hamburgers and 0.99 for salami.

Table 13: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from RawFoot dataset) with supervised learning classifiers. * denotes highest accuracy achieved.

RawFoot Dataset - Training/Testing Split								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3 pool**			
	Acc (%)	Recall	F1	Kappa	Acc (%)	Recall	F1	Kappa
GNB	82.02	0.82	0.83	0.82	78.42	0.78	0.79	0.78
SVM-RBF	99.10	0.99	0.99	0.99	96.63	0.97	0.97	0.97

ACCEPTED MANUSCRIPT

SVM-Poly	98.21	0.98	0.98	0.98	96.74	0.97	0.97	0.97
ANN	99.28*	0.99	0.99	0.99	97.04	0.97	0.97	0.97
RF	98.13	0.98	0.98	0.98	94.03	0.94	0.94	0.94

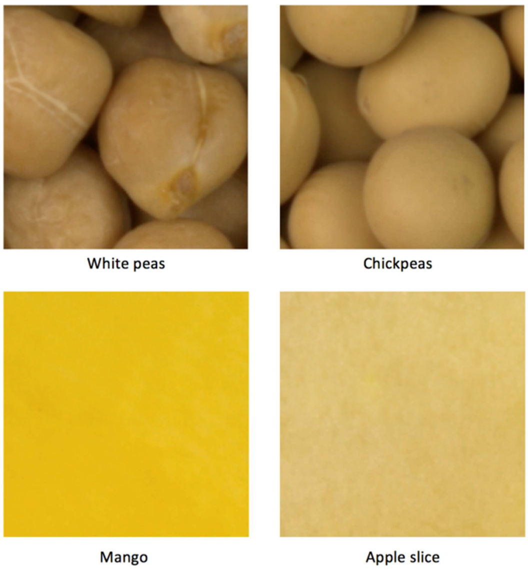


Figure 8: Example of RawFooT-DB classes which are misclassified based on confusion matrix generated from SVM-RBF model trained using ResNet-152 features. Images highlight shared characteristics that could lead to misclassifications.

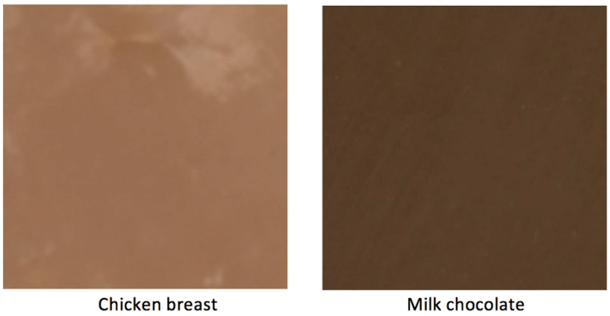


Figure 9: Example of RawFooT-DB classes which are misclassified based on confusion matrix generated from ANN model trained using ResNet-152 features.

For further analysis using RawFooT-DB with ResNet-152 and GoogleNet features, we reordered the food types into 7 groups, vegetables, rice/grains/wheat/seeds, fruits, sweets, breads, meat/fish, and miscellaneous (e.g. coffee, powders, sugar). Figure 10 and 11 show the F-measure of the food texture types rearranged into food groups for ANN and SVM-RBF models. It is clear the from Figure 10 and11 that there is a decrease in accuracy in ‘meat/ fish’ group. This is evident in Figure 9 as chicken breast can share similar characteristics with other textures such as ’milk chocolate’. Figure 10 and 11 also show decrease in accuracy with chickpeas and white peas due to sharing texture and shape characteristics and this is also evident in Figure 12 using GoogleNet deep features with ANN.

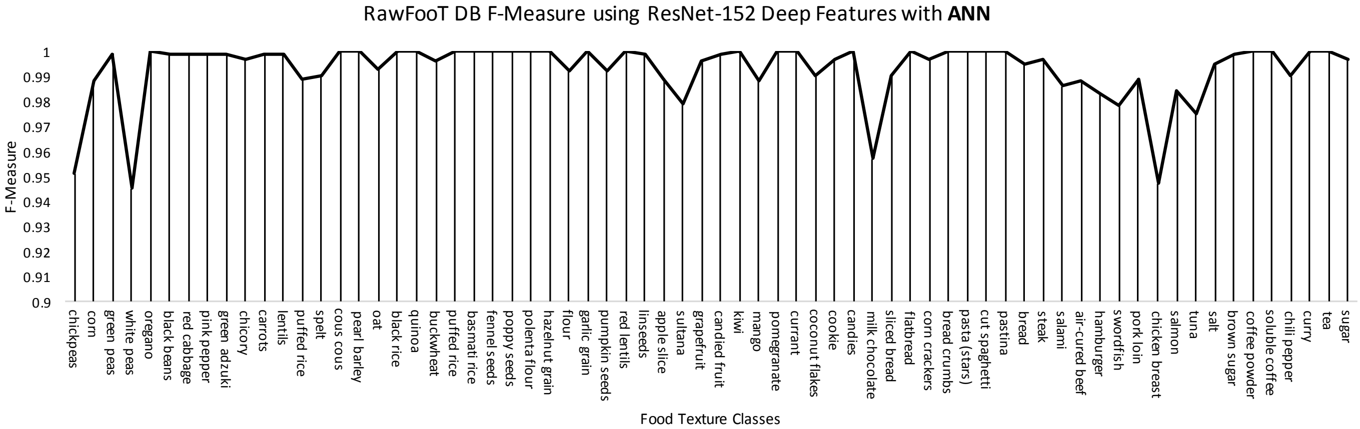


Figure 10: RawFooT-DB F-Measure of reordered classes by major food groups using ResNet-152 features with ANN.

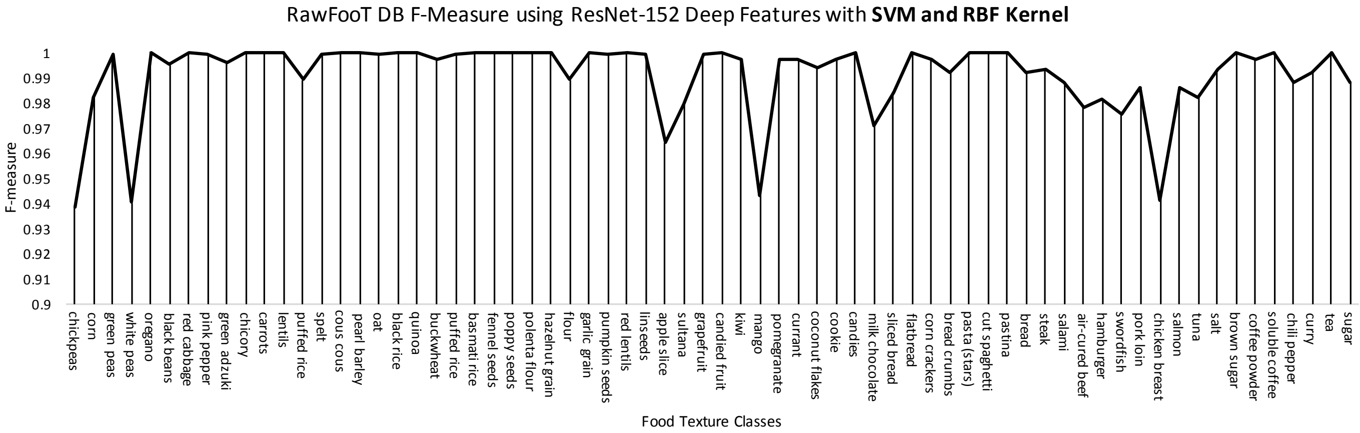


Figure 11: RawFooT-DB F-Measure of reordered classes by major food groups using ResNet- 152 features with SVM with RBF kernel.

ACCEPTED MANUSCRIPT

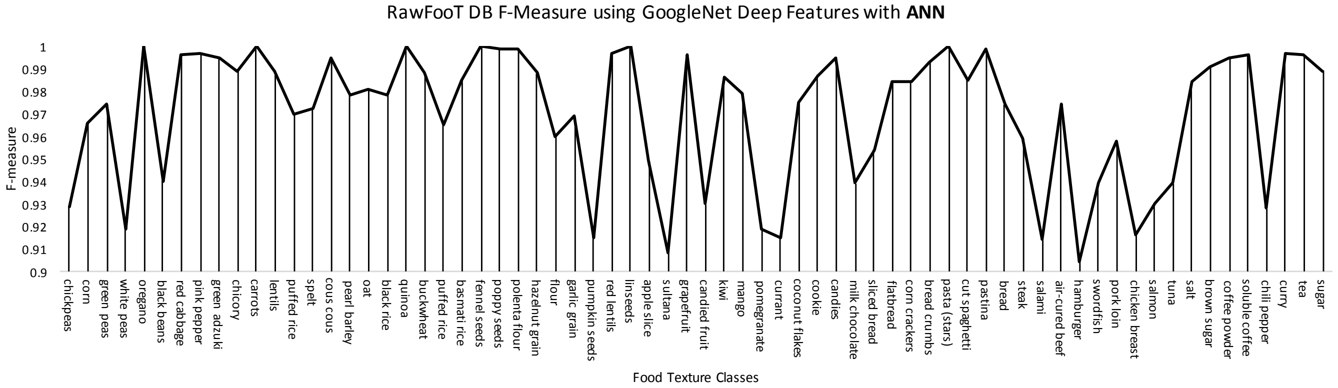


Figure 12: RawFooT-DB F-Measure of reordered classes by major food groups using GoogleNet features with ANN.

5.2.3. Food-101 Classification Results

From previous experiments using Food-5K and Food-11, and RawFooT-DB, ResNet-152 deep features achieved the highest accuracies. We used ResNet-152 deep features for classifying Food-101, which can be described as fine-grained food image dataset that contains similar food items (i.e. different kind of soups, meats images taken in a free-living environment). Results listed in Table 14 show that ANN and SVM-RBF along with ResNet-152 features achieved the highest accuracy across the experiments for Food-101 achieving 64.98%. To train the ANN, Food-101 was partitioned into 75:25, training and testing, with random seed of ’1’ using Weka 3.8.1 (same ANN plug-in used with other experiments for Food-5K, Food-11, and RawFooT-DB). To train the ANN, the learning rate was initially set to 1 with mini-batch gradient descent. For the other classification models we used used Python 2.7.10 with Scikit v0.19. We used Python v2.7.10 and scikit-learn instead of Weka 3.8.1 due to the flexibility of using other libraries and its ease of use when working with larger datasets and also for data analysis. The parameters for the classifiers remained the same as other experiments with Weka as wekaPython contains the same models as scikit-learn. To train the other classifiers using scikit-learn, Food-101 was also split in 75:25 training and testing with a random state parameter of ’1’. Table 14 shows the accuracy, recall, F-Measure, and kappa statistic of using ResNet-152 deep features. The results are much lower than previous experiments with the highest accuracy with 64.18% for ANN and 64.97% for SVM-RBF. The kappa statistic was also generated for ANN and SVM-RBF at 0.64 and 0.65 respectively, which indicates substantial agreement.

Table 14: Classification results using ResNet-152 to extract deep activations (extracted from Food-101 dataset)

with supervised learning algorithms. Highest accuracy denoted by *.

Food-101 Dataset - 75:25 training/evaluation				
Model	ResNet-152 - pool5			
	Acc (%)	Recall	F1	Kappa
GNB	45.64%	0.46	0.46	0.45
SVM-RBF	64.98%*	0.65	0.65	0.65
SVM-Poly	63.04%	0.63	0.63	0.63
ANN	64.18%	0.64	0.64	0.64
RF	39.33%	0.39	0.38	0.39

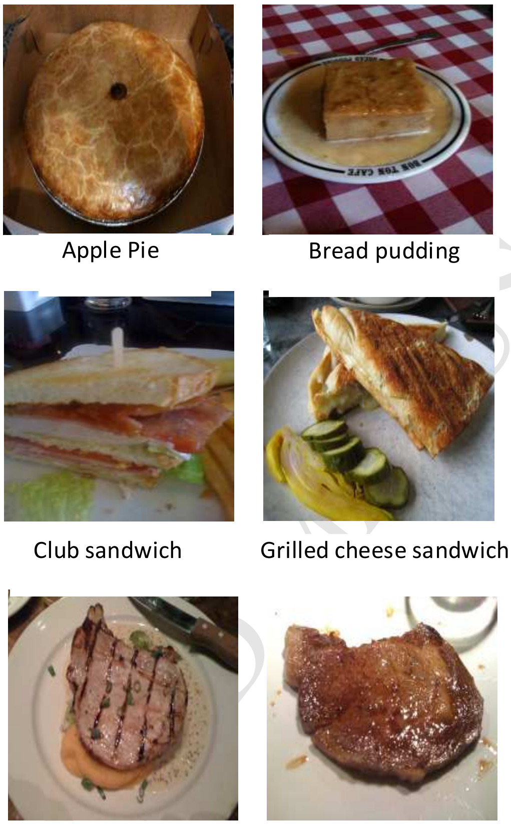
There were a number of misclassifications that occurred across different classes in Food-101 experiments. Figure 13 and 14 is an example of typical food classes that were misclassified. Misclassifications occured with the steak food class with both the ANN and SVM-RBF. Steak instances were wrongly classified as pork chop, prime rib, and filet mignon using SVM-RBF and ANN, similarly several pork chop instances were classified as steak, prime rib, and foie gras. This may be due to the shared characteristics with shape, texture, and colour. In regards to the desserts, several items were wrongly classified, the panna cotta class was wrongly classified as a cheese cake, and chocolate mousse and the cheese cake class was wrongly classified as a panna cotta, choco- late mousse, chocolate cake, and strawberry shortbread. Further investigation showed that these classes share similar characteristics such as shape and colour which may contribute to them being wrongly classified. Beignets were also wrongly classified as donuts, investigation showed that beignets are very similar to donuts in terms of appearance, texture, colour, and shape, however SVM- RBF trained with ResNet-152 features were still able to achieve an F-measure of 0.77 for beignets.

Figure 15 shows the F-measure for each food class in Food-101 for SVM. For further analysis, we organised the food classes into groups. Images were allo- cated into groups; (1) breads, pasta, (2) desserts, (3) eggs, (4) fried foods, (5) meats and fish, (6) mixed foods (foods that contained a mixture of foods) and (7) vegetables. Foods were organised into different foods to determine if ResNet-152 features had any inherent advantage for classifying certain food groups. The average F-measure was computed for each group and the vegetable group achieved the highest with an average F-measure of 0.71 using SVM-RBF model, however it should be noted that the vegetable category contained a small number of images in comparison to other groups. In regard to using SVM-RBF model to classify specific food items, the class the achieved the highest F-measure was ’edamame’ with 0.98, and further investigation showed that edamame images are very similar as the food item is distinct and there is little variation with the edamame food type

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

and also they are the same shape and colour. The food item that achieved the lowest F-measure was ’steak’ with an F-measure of 0.36. Steak food class experienced misclassifications with other food types with other meat classes e.g. pork chop, prime rib, and foie gras due to the similar shape, colour, and texture. In regards to using ANN model, ’edamame’ also achieved the highest with 0.97 F-measure and ’steak’ was also the lowest with 0.30.



Apple Pie

Bread pudding

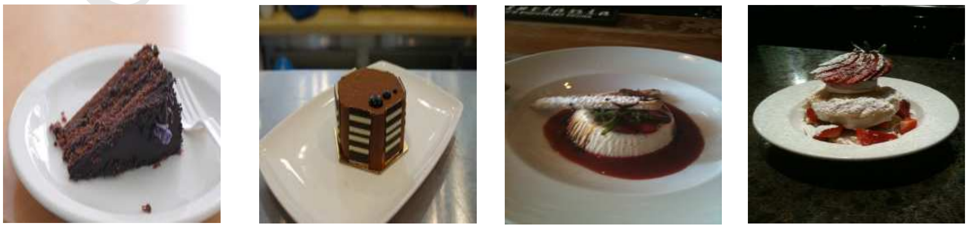
Club sandwich

Grilled cheese sandwich

Pork chop

Steak

Figure 13: Example of Food-101 classes which were misclassified based on confusion matrix generated from ANN and SVM-RBF models trained using ResNet-152 features. Food classes are on the left experience misclassification with the food classes on the right.



chocolate cake

chocolate mousse

panna cotta

strawberry shortbread

Figure 14: Example of Food-101 dessert classes which were misclassified based on confusion matrix generated

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

using both SVM-RBF and ANN models trained with ResNet-152 features.

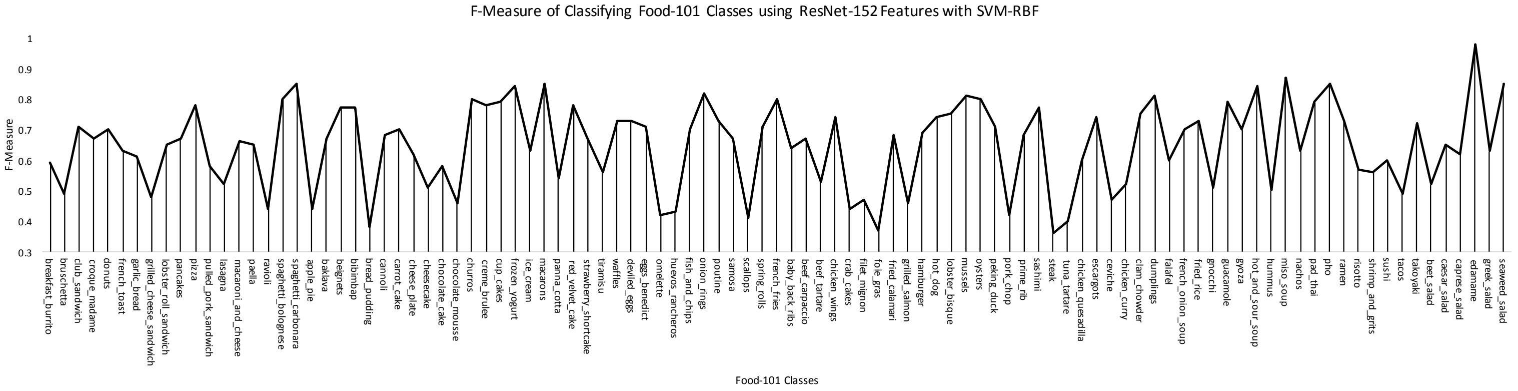


Figure 15: Food-101 F-Measure of reordered classes by major food groups using ResNet-152 features with SVM with RBF kernel

6. Discussion

In this work we used deep features extracted from pretrained CNNs for food image classification. We compared 2 popular pretrained CNNs, ResNet-152 and GoogLeNet and extracted deep features from layers deep in each CNN architec- ture to classify Food-5K, Food-11, and RawFooT-DB. For Food-101 we choose to use ResNet-152 deep features as it consistently achieved higher accuracies across other image datasets. We extracted a deep feature vector immediately after the last pooling layer in each architecture for each pretrained CNN for each from various food image datasets. From these experiments, we found that ResNet- 152 achieved consistently higher results in Food-5K, Food-11, and RawFoot-DB and because of this ResNet-152 features were used with Food-101. Food-101 is a much more difficult dataset due to the number of classes and variation in images. Many classes contain low in between class variance as many dishes are similar as shown in Figure 13, 14, and 16. From the experiments it was clear that using ResNet-152 is able to achieve high accuracies for Food-5K, Food-11 dataset, RawFoot DB, and moderate accuracy for Food-101.

In regards to Food-5K, the deep features were able to detect food in images with high accuracy across all machine learning classifiers, achieving over 90% accuracy in each experiment. We benchmarked our experiments using the results achieved by the authors of Food-5K and Food-11 datasets who used a fine-tuned GoogleNet [13] and these results in our work suggest that there is potential to achieve high accuracies and performance without the need of fine- tuning pretrained CNNs for certain datasets and problems. Furthermore, due to the nature of Food-5K being a binary decision between food and non-food classes, generic deep features may be sufficient enough to provide adequate generalisation to classify between two classes (i.e. food and non-food).

ANN and SVM-RBF trained with ResNet-152 features achieved the highest accuracies in the majority of Food-5K experiments and the Food-5K ANN and SVM-RBF model was further evaluated by classifying the entire Food-11 dataset for food detection. Results show that our

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

ANN model trained using ResNet-152 features achieved higher food detection accuracy compared to the fine-tuned GoogleNet model in [13] when tested against Food-11 image dataset as stated in Table 15. We also evaluated both our Food/Non-Food SVM-RBF model trained with ResNet-152 and GoogleNet deep features using Food-11 for food detection and results showed that these models achieve marginally higher results compared to other results achieved in also listed in Table 15 [13].

Authors in [13] achieved 83.6% with Food-11 evaluation dataset and in our work ResNet-152 features with ANN achieved 91.34% and 89.99% with SVM-RBF, this is an improvement of 7.74% and 6.39% respectively. For Food-5K, ResNet-152 features achieved 98.8% in classifying Food-5K evaluation dataset and authors in [13] achieved 99.2%. Authors in [13] evaluated their food detection model using all images in Food-11 dataset, we did this also and Table 16 compares our results. ANN and SVM trained with ResNet-152 deep features achieved marginally higher results than achieved in [13] with 97.39% and 97.19% respectively. GoogleNet deep features with ANN also achieved marginally higher results with 97.16% compared to proposed Fine-tuned GoogleNet method in [13].

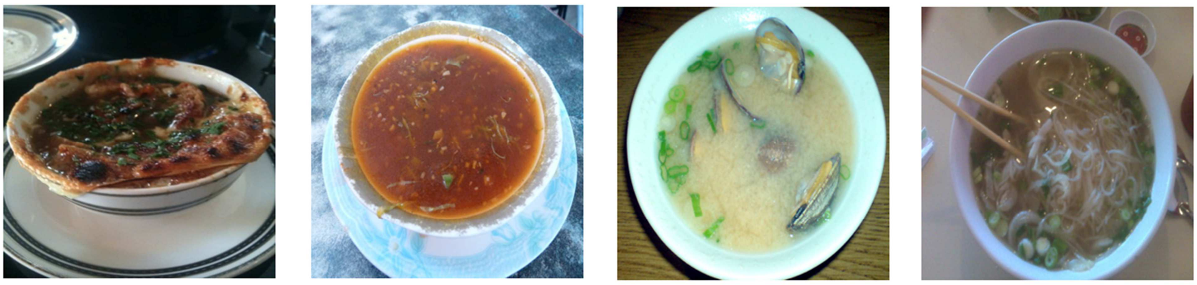


Figure 16: Food image classes from Food-101 that share similar characteristics. Categories from left to right; french onion soup, hot and sour soup, clam chowder, miso soup.

Table 15: Method and results comparison using Food-5K and Food-11. * denotes accuracy improvement.

Author	Method	Accuracy	Food Dataset
Singla, et al. [13]	GoogleNet (fine-tuned)	99.2%	Food-5K
Singla, et al. [13]	GoogleNet (fine-tuned)	83.6%	Food-11
This work	ResNet-152 + ANN	98.8%	Food-5K
-	ResNet-152 + ANN	91.34%*	Food-11
-	ResNet-152 + SVM-RBF	89.99%*	Food-11

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

-	ResNet-152 + SVM-Poly	88.86%*	Food-11

Table 16 also shows GoogleNet features used to detect food images in Food- 11. Results show that using GoogleNet features used to train conventional machine learning algorithms is able to achieve higher results than a fine-tuned GoogleNet model in detecting food images in Food-11. These results illustrate the convenience of using deep learning with machine learning classifiers through deep feature extraction as the user does not need to use a powerful GPU to quickly train an effective image classification model. Many deep learning pack- ages such as Tensorflow and MatConvNet give users the ability to fine-tune CNNs using CPU, however it has been stated that using a GPU can be around 8 times faster than using a CPU in training a CNN [40].

Table 16: Results comparison of classifying Food-11 with our Food/Non-Food classification models. * denotes accuracy improvement.

Method	**Number of Food Images Detected**	**Accuracy**
GoogleNet (fine-tuned) [13]	16,127	96.9%
ResNet-152 + ANN	16,208	97.39%*
ResNet-152 + ANN	16,176	97.19%*
ResNet-152 + SVM-RBF	16,171	97.16%*
ResNet-152 + SVM-Poly	15,646	94.00%

Food-5K AUC results achieved in this work were close to 1 in validation and evaluation image sets using ANN and RF with both ResNet-152 features and GoogleNet features. However, the validation and evaluation test sets are small in comparison to other popular food image datasets with only 500 in each class for each dataset and therefore more research is needed in classifying a wider range of food images types and image quality. Food-5K training dataset, which was used to train food/non-food models, is also comparatively small with 2500 images in each class and contains limited food image types, therefore further re- search would need to be completed in training machine learning classifiers with a diverse food image training dataset. Further evaluation was completed using the food/non-food trained models that achieved highest accuracies with Food- 5K to classify a new image dataset that combines food images in UNICT-FD889 and non-food images Caltech-101, called UNICT-Caltech, which is larger than the validation and evaluation sets provided in Food-5K [52, 53] containing 3583 food images and 9144 non-food images . Results from classifying this dataset are listed in Table 10 and show that with using Food-5K training dataset to train machine learning classifiers is able to achieve a high food accuracy using SVM-RBF achieving 97.50%.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Further experiments focused on using deep features to classify food texture image items under different illuminations, previous authors of RawFooT DB re- searched the use of using other popular pretrained CNNs for feature extraction. The experiments presented in this work utilised deep residual network features and GoogleNet features to classify food images in different lighting settings. Other research that used RawFooT-DB [20] divided the food image classes into illuminant categories. In this work, we evaluated the performance of ResNet-152 features in classifying food texture images across a range of different lighting conditions. Results from using ResNet-152 to train an ANN achieved 99.28% accuracy and and a ROC value of 0.99 and the same features with SVM-RBF achieved 99.10%. More importantly, the use of deep features with supervised machine learning algorithms, from both ResNet-152 and GoogLeNet, are able to generalise between food texture types with great efficiency under different illuminations. Results from RawFooT-DB echos results in early experiments in that ResNet-152 features marginally outperform GoogleNet features even in de- termining food classes across a number of illuminations. Figure 12 highlights the performance of classifying each texture class in RawFooT-DB using GoogleNet features with ANN, and similar decreases in F-measures are present when com- pared to ResNet-152 ANN and SVM-RBF in Figure 10 and 11. GoogleNet features also experienced misclassifications with white peas and chick peas, and with several meat textures (salami and hamburger).

Results show that most experiments with RawFooT-DB using both feature types achieved over 90% accuracy (apart from GoogleNet features with Gaussian Naive Bayes, which achieved 78.42%), however ResNet-152 pretrained CNN features achieves higher accuracy across all machine learning algorithms. This may be due to the increased depth of ResNet-152 CNN in comparison to GoogLeNet CNN and therefore rich detailed features may be extracted from layers deep in ResNet architecture. Pretrained CNN models used in this work were supplied by MatConvNet and experiments in [58] show that ImageNet ILSVRC trained ResNet-152 model outperformed ImageNet ILSVRC trained GoogLeNet Inception model when validating both using ImageNet ILSVRC 2012 validation data using MatConvNet package [58].

There were also several misclassifications between similar food groups with RawFooT-DB. It is worth noting that these food textures that were misclassified are very alike in texture and shape (chickpeas and white peas) and the images used for testing and training are focused on the food texture without the overall food item shape and size as shown in Figure 8 and 9. The use of a texture based classification model trained using deep features may also be very efficient combined with a semi-automation approach to food logging. Future work could enable the user to utilise a polygonal tool to draw around the food item and then a food texture based classifier can you used to predict the food item thus removing much of the complexity and noise of other food and non-food items in the food image. It is clear from the experiments that using pretrained ResNet CNN for deep feature extraction is able to produce feature descriptors that generalise accurately between food texture classes with low in-between variance.

It was revealed that ResNet-152 features continually achieved higher classification accuracy results when compared to GoogleNet therefore ResNet-152 deep features were used to classify Food-101 dataset. The images in Food-101 were not developed in a controlled environment but collated using a social media website (Foodspotting), which were uploaded by users and taken in

ACCEPTED MANUSCRIPT

real world environments (restaurants, at home, cafes, etc.).The images are also taken under illuminations and the dataset contains image quality of the images vary greatly and no bounding box information is provided to help determine where the food items are located in the image. Food-101 contains 101,000 images and 1,000 for each food class, and because of the size of this dataset, we partitioned dataset in training and validation using 75:25 ratio, 75% used for training and 25% used for testing and used a random state of ’1’ with scikit-learn library. The highest accuracy achieved using ResNet-152 deep features extracted from Food-101 was 64.98% using an SVM with RBF kernel using ResNet-152 features. The full breakdown of results using ResNet-152 to classify Food-101 are located in Table 14. The features extracted from layers deep in CNN architecture pro- vide efficient representations that can be used to classify even the most difficult food image datasets such as Food-101. The quality of food images present in Food-101, in regards to food variation and noise i.e. other non-food items, and unrelated food items, may be a factor in the decrease in accuracy. Comparing the results of Food-101 (101 classes) with RawFooT-DB texture dataset (67 classes) suggest that the class size may not a major determining factor in the decrease in accuracy but the quality of the images used in regards to being truly representative of the class. Results achieved in this work in classifying RawFooT-DB is comparable with results achieved in [20] albeit the authors created small subsets for each lighting condition, while work presented in this paper extracted features from each food class that contains a variety of lighting conditions.

For further comparison, Table 17 lists results achieved in this work with other research that used related deep feature extraction in classifying food image datasets. It is clear from Table 17 and the literature that ResNet-152 deep features echo results achieved with other datasets and other deep feature types [45]. ResNet-152 deep features are able to achieve high classification accuracy in both fine grained datasets such as RawFooT-DB and binary decision datasets e.g. Food/NonFood, however there is a decrease in accuracy when food image datasets with high food variance and noise is present in images as seen in Food- 101. A semi-automated approach or segmentation approach could be applied to CNN deep feature classification that allows the user to draw around a food image before classification to remove noise, further analysis is needed to evaluate this approach and to measure improvement in accuracy.

Table 17: Summary of research using deep feature extraction and fine-tuning methods to classify various food image datasets. **Bold** denotes results achieved in this work. * denotes highest accuracy achieved for Food-5K, Food-11, and RawFooT-DB.

Extraction Model	**Accuracy**	**Food Classes**	**Food Dataset**
VGG-S [41] NIN AlexNet	92.47% 90.82% 84.95%	2 (Food/Non-Food) 2 (Food/Non-Food) 2 (Food/Non-Food)	RagusaDB
GoogleNet [42]	94.67% 99.01%	2 (Food/Non-Food) 2 (Food/Non-Food)	Based on RagusaDB FCD

ACCEPTED MANUSCRIPT

NIN [47]	95.1%	2 (Food/Non-Food)	IFD
GoogleNet [13]	99.2%* 83.6%	2 (Food/Non-Food) 11	Food-5K (Evaluation dataset) Food-11 (Evaluation dataset)
AlexNet [15]	94.01% 70.13%	7 (Food groups) 61	PFID PFID
AlexNet [45] AlexNet [45] AlexNet [45] AlexNet [45]	57.87% 70.41% 78.77% 67.57%	100 101 100 256	UEC-FOOD100 Food-101 UEC-FOOD100 UEC-FOOD256
VGG-19 [46] Overfeat-Fast [46]	40.21% 33.91%	101 101	ETHZ-Food-101
VGG-16 [57] VGG-19 [57]	98.21% 97.69%	68 68	RawFooT-DB RawFooT-DB
ResNet-152 + ANN **ResNet-152+ ANN** **ResNet-152 + ANN** **ResNet-152 + ANN** **ResNet-152 + SVM-RBF**	**98.8%** **99.4%** **91.34%*** **99.28%*** **64.98%**	**2 (Food/Non-Food)** **2 (Food/Non-Food)** **11** **68** **101**	**Food-5K (Evaluation dataset)** **Food-5K (Validation dataset)** **Food-11 (Evaluation dataset)** **RawFooT-DB (testing dataset)** **Food-101**

Using CNN deep features to classify food images datasets exceed the performance compared to other conventional feature selection methods and has been well documented [45,49,51]. Hand crafted feature selection methods such as SURF, or colour can encounter difficulties when classifying fine-grained classification of food categories as some public food image datasets contain small in-between class differences amongst large number of classes (e.g. Food-101). It has been stated in [51] that deep CNN features should be the first initial method for visual classification tasks due to their high performance in generalising to other datasets as CNNs are trained to be able to learn rich representations from a large number of images. CNNs able to determine complex filters to combine them with other patterns for greater detail. CNNs are able to produce internal image feature representation, which is advantageous when compared to hand crafted feature types such as SIFT, SURF or HOG. In this work, ResNet-152 features are able discriminate effectively between food and non-classes and in classifying high level food groups (Food-11), when compared to other works in [13]. It is clear that using ResNet-152 pretrained model is able to capture relevant image features to enhance the generalisation between fine-grained objects as demonstrated in classifying RawFooT DB in table . ResNet-152 contains 152

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

layers that combine multiple convolutional and pooling layers to filter important image features and the use of residual connections to train the network produce accurate features which can be highlighted for effective generalisation across other datasets.

It is clear that using CNN features can enhance the accuracy of food image classification when compared to traditional feature extraction methods and this has been observed in other works, for example in [17] SURF and LAB colour features, and Random Forests were used to classify Food-101 dataset and achieved 50.76% accuracy. In [45] an AlexNet model was fine-tuned using food image categories and deep feature extraction was performed after to classify Food-101, and authors achieved 70.41%, which is a significant increase when compared to results achieved in [17]. As well as deep feature extraction, fine-tuning was also used to classify Food-101 and authors in [48] achieved top-1 accuracy of 77.4% after 250,000 iterations in training a CNN architecture called ’DeepFood’, which is a significant accuracy increase in comparison to [17]. In [49] fine-tuning was also used to classify Food-101 dataset was also used to fine-tune Inception V3 architecture and achieved a top-1 accuracy of 88.28%. Research in [45] also achieved a top-1 accuracy of 65.32% using HOG features, colour values with fisher vectors in classifying UEC-FOOD100, however CNN based features extracted from a modified AlexNet model with a linear SVM achieved an in- creased accuracy of 78.77%. For UEC-FOOD256 dataset, work presented in [50] achieved a top 1 accuracy of 50.1% using HOG features and colour features with Fisher Vector representations and the same authors in later research [45] utilise deep CNN features extracted from a modified AlexNet and achieved a top 1 accuracy of 67.57% in also classifying UEC-FOOD256 dataset. For RawFooT-DB food texture dataset experiments were completed in classifying food textures under various lighting conditions, authors compared traditional feature extraction techniques with CNN based features, and results show that OCLBP and Gabor features achieved 95.9% and 96.2% accuracy respectively with deep CNN features achieving 98.2% accuracy [20]. From the literature it is clear that using CNN deep feature extraction and fine-tuning can achieve superior results in regards to food image classification.

7. Limitations & Future Work

There are a number of limitations associated with this study which could be addressed in future works, for example, an expansive dataset could be developed under a controlled environment that is representative of a broad range of food items. This dataset could be used with the methods outlined in this work and compared with similar works. This would give a clear indication of the true performance of using deep feature extraction with machine learning algorithms. Also, a comprehensive study could be completed by fine-tuning a range of CNNs on food datasets and comparing performance using the same pre- trained CNN models for deep feature extraction. Further experiments can also be completed by comparing deep features extracted from different layers within a CNN architecture to find what layer is more suitable for generalising between different food classes. In regards to overfitting, particularly for Food-101, future works could include using 10-fold class validation instead of using a 75:25 train/testing split. This would give a clearer indication of the performance of using deep features from ResNet-152 and GoogLeNet. Some of the experiments in this work achieved high accuracies, especially for Food/Non-Food classification experiments, however it is important to note that the amount of images contained in Food-5K are relatively small in comparison to other datasets e.g. Food-11 or Food-101.

ACCEPTED MANUSCRIPT

Further experiments need to be completed in detecting food/non-food in larger food image datasets in using off the shelf deep features.

For RawFoot-DB we used the training and test split provided by authors in [20, 42], however the authors of RawFooT DB in [20] created subsets of each category, which were based on lighting condition type. In this work, our aim was to classify food textures across different lighting conditions, however in future work we would follow the same procedures described in [20] and use ResNet-152 features for further comparison. Also authors of [17] allocated a testing split that contained images that contained little noise and representative of each class, however in our work Food-101 extracted features were shuffled using random seed ’1’ and random state ’1’ to determine the classification performance of ResNet-152 features when used with images with high level of noise. In future works, we will further evaluate ResNet-152 features following the partition procedure described in [17].

Future work could incorporate hierarchical classification using pretrained CNN features in which a classifier will be used to determine food and non-food images, another classifier will be appended that determines major food groups, and finally a further classifier will used after to determine low level food item. Further experiments with the parameters of machine learning models could also be changed in order to determine the optimal parameter settings to achieve a high classification accuracy. The presence of noise in the food image datasets may also affect the accuracy, in order to mitigate these issues, a semi-automated approach could be adopted by using a polygonal tool to draw around the food portion and to ultimately segment the food item. Classification models could then classify the segmented food portion in order to promote accuracy. Other computer vision segmentation approaches could be researched and combined with methods described in this work. For future evaluation, we would also in- put random noise as feature vectors for trained classifiers to determine food classes and analyse the output and performance. The use of machine learning models using pretrained CNN deep features also have the potential of being using in mobile health solutions. Much research has been dedicated to under- standing a person’s diet by determining what major food groups they consume daily [2,5]. This research has showed that this process can be automated using deep features extracted from residual CNNs for high food classification accuracy. From this research, it is clear that ResNet-152 deep features is able to distinguish between high-level food categories such as Food/Non-food and echoes other related research in this area. In comparison with other works, ResNet- 152 deep features outperforms other CNN deep features such as GoogleNet in distinguishing between fine-grained food texture classes in RawFooT DB and is comparable with other related works [20]. ResNet-152 features encountered some difficulty in classifying Food-101 classes, however this may be due to the images containing noise in the form of high colour intensities and multiple foods in the same image, however a reasonable accuracy of 64.98% was achieved. In Food-11 food group classification, deep GoogleNet features were able to achieve high accuracy result when compared to research presented in [13] which used a fine-tuned GoogleNet, which shows that a combination of conventional ma- chine learning classifiers combined with CNN deep features have the ability to outperform fine-tuned models.

8. Acknowledgements

ACCEPTED MANUSCRIPT

Thank you to Department of Education and Learning Northern Ireland (DELNI) for funding this research.

9. References

1. M. Di Cesare et al., “Trends in adult body-mass index in 200 countries from 1975 to 2014: A pooled analysis of 1698 population-based measure- ment studies with 19.2 million participants,” The Lancet, vol. 387, no. 10026. pp. 1377-1396, 2016.

2. Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, “Food image analysis: Segmentation, identification and weight estimation,” Proc. - IEEE Int. Conf. Multimed. Expo, 2013.

3. T. Lehnert, D. Sonntag, A. Konnopka, S. Riedel-Heller, and H.-H. Knig, “Economic costs of overweight and obesity,” Best Pract. Res. Clin. En- docrinol. Metab., vol. 27, no. 2, pp. 105-115, 2013.

4. C. M. Wharton, C. S. Johnston, B. K. Cunningham, and D. Sterner, “Dietary Self-Monitoring, But Not Dietary Quality, Improves With Use of Smartphone App Technology in an 8-Week Weight Loss Trial,” J. Nutr. Educ. Behav., vol. 46, no. 5, pp. 440-444, 2014.

5. M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, “A food recognition system for diabetic patients based on an optimized bag-of-features model,” IEEE J. Biomed. Heal. Informatics, vol. 18, no. 4, pp. 1261-1271, 2014.

6. G. M. Farinella, M. Moltisanti, and S. Battiato, “Food recognition using consensus vocabularies,” in Lecture Notes in Computer Science (includ- ing subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9281, pp. 384392.

7. H. He, F. Kong, and J. Tan, “DietCam: Multiview food recognition using a multikernel SVM,” IEEE J. Biomed. Heal. Informatics, vol. 20, no. 3, pp. 848-855, 2016.

8. N. Martinel, C. Piciarelli, C. Micheloni, and G. L. Foresti, “A Struc- tured Committee for Food Recognition,” in Proceedings of the IEEE In- ternational Conference on Computer Vision, 2015, vol. 2015February, pp. 484-492.

9. ImageNetLargeScaleVisualRecognitionCompetition(ILSVRC)”,Image- net.org, 2017. [Online].
Available: http://www.image-net.org/challenges/LSVRC/. [Accessed: 16- Sep- 2017].

10. A. Krizhevsky, I. Sutskever, and G. E. Hinton,“ImageNet Classification with Deep Convolutional Neural Networks,” Adv. Neural Inf. Process. Syst., pp. 19, 2012.

11. N. Tajbakhsh et al., “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?,” IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 12991312, 2016.

ACCEPTED MANUSCRIPT

12. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Work- shops, 2014, pp. 512519.

13. A. Singla, L. Yuan, and T. Ebrahimi, “Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model,” in Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa 16, 2016, pp. 311.

14. H. , K. Aizawa, and M. Ogawa, “Food Detection and Recognition Using Convolutional Neural Network,” ACM Multimed., no. 2, pp. 10851088, 2014.

15. M. Farooq and E. Sazonov, “Feature Extraction Using Deep Learning for Food Type Recognition” in Bioinformatics and Biomedical Engineering: 5th International Work-Conference, IWBBIO 2017, Granada, Spain, April 26–28, 2017, Proceedings, Part I, I. Rojas and F. Ortuo, Eds. Cham: Springer International Publishing, 2017, pp. 464-472.

16. Y. Kawano and K. Yanai,“Food Image Recognition with Deep Convolutional Features” ACM Int. Jt. Conf. Pervasive Ubiquitous Comput., pp. 589-593, 2014.

17. Bossard, M. Guillaumin, and L. Van Gool, “Food-101 - Mining discriminative components with random forests,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8694 LNCS, no. PART 6, pp. 446-461.

18. Y. Matsuda, H. Hoashi, and K. Yanai,“Recognition of multiple-food im- ages by detecting candidate regions” in Proceedings - IEEE International Conference on Multimedia and Expo, 2012, pp. 2530.

19. Y. Kawano and K. Yanai,“Automatic expansion of a food image dataset leveraging existing categories with domain adaptation,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intel- ligence and Lecture Notes in Bioinformatics), 2015, vol. 8927, pp. 317.

20. C. Cusano, P. Napoletano, and R. Schettini, “Local angular patterns for color texture classification,” in Lecture Notes in Computer Science (includ- ing subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9281, pp. 111-118.

21. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, “Large-scale video classification with convolutional neural networks,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732.

22. C. Szegedy et al., “Going deeper with convolutions,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 7-12-2015, pp. 19.

ACCEPTED MANUSCRIPT

23. A. Vedaldi and K. Lenc, MatConvNet, in Proceedings of the 23rd ACM international conference on Multimedia - MM 15, 2015, pp. 689-692.

24. ”Image Category Classification Using Deep Learning”, Mathworks.com, 2017. [Online]. Available: https://www.mathworks.com/examples/matlab-computer-vision/mw/vision product- DeepLearningImageClassificationExample-image-category-classification-using- deep-learning. [Accessed: 18- Sep- 2017].

25. K. He, X. Zhang, S. Ren, and J. Sun,“Deep Residual Learning for Image Recognition” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

26. K. Simonyan and A. Zisserman,“Very Deep Convolutional Networks for Large-Scale Image Recognition,” Int. Conf. Learn. Represent., pp. 114, 2015.

27. 28. R. G. Pontius and M. Millones, “Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment,” Int. J. Remote Sens., vol. 32, no. 15, pp. 4407-4429, 2011.

28. “Weka 3 - Data Mining with Open Source Machine Learning Software in Java,” Cs.waikato.ac.nz, 2017. [Online].
Available: http://www.cs.waikato.ac.nz/ml/weka/index.html. [Accessed: 18- Sep- 2017].

29. “Waikato Environment for Knowledge Analysis(WEKA),”Weka.sourceforge.net, 2017. [Online].
Available: http://weka.sourceforge.net/packageMetaData/wekaPython/Latest.html. [Accessed: 18- Sep- 2017].

30. “Java (convolutional or fully-connected) neural network implementation,” GitHub, 2017. [Online].
Available: https://github.com/amten/NeuralNetwork/releases/tag/v1.1. [Accessed: 18- Sep- 2017].

31. H. Zhang,“The Optimality of Naive Bayes,” Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004, vol. 1, no. 2, pp. 16, 2004.

32. I. H. Witten, E. Frank, and M. a Hall, Data Mining: Practical Machine Learning Tools and Techniques. 2011.

33. T. Malisiewicz, A. Gupta, and A. A. Efros,“Ensemble of exemplar-SVMs for object detection and beyond,” in Proceedings of the IEEE Interna- tional Conference on Computer Vision, 2011, pp. 89-96.

34. C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass sup- port vector machines,” IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 415-425, 2002.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

35. G. James, D. Witten, T. Hastie, and R. Tibishirani, An Introduction to Statistical Learning. 2013.
36. J. P. Mueller, et al,“Hitting Complexity with Neural Networks,” Machine Learning for Dummies, Hoboken, New Jersey, Wiley, 2016, ch. 16, pp.279- 290.

37. L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 532, 2001

38. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346359, 2008.

39. T. Ojala, M. Pietikinen, and T. Menp, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971987, 2002.

40. “GPU vs CPU in Convolutional Neural Networks using TensorFlow — Relink”, Relink, 2017. [Online]. Available: https://relinklabs.com/gpu- vs-cpu-in-convolutional-neural-networks-using-tensorflow. [Accessed: 19- Sep- 2017].

41. H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, Automatic diet monitoring: a review of computer vision and wearable sensor-based methods., Int. J. Food Sci. Nutr., pp. 115, 2017.

42. “RawFooT DB”, Projects.ivl.disco.unimib.it, 2017. [Online]. Available: http://projects.ivl.disco.unimib.it/minisites/rawfoot//. [Accessed: 19- Sep- 2017].

43. F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella, Food vs Non-Food Classification, in Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa 16, 2016, pp. 77-81.

44. E. Aguilar, et al. ”Exploring Food Detection using CNNs,”arXiv:1709.04800v1 [cs], Sept 2017.

45. K. Yanai and Y. Kawano, Food image recognition using deep convolu- tional network with pre-training and fine-tuning, in 2015 IEEE Interna- tional Conference on Multimedia & Expo Workshops (ICMEW), 2015, pp. 1-6

46. X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, Recipe recog- nition with large multimodal food dataset, in 2015 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2015, 2015.

47. Kagaya and K. Aizawa, Highly Accurate Food/Non-Food Image Classifi- cation Based on a Deep Convolutional Neural Network BT - New Trends in Image Analysis and Processing – ICIAP 2015 Workshops: ICIAP 2015 International Workshops, BioFor, CTMR, RHEUMA, ISCA, MADiMa, SBMI, and QoEM, Genoa, Italy, September 7-8, 2015, Proceedings, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Cham: Springer International Publishing, 2015, pp. 350-357.

ACCEPTED MANUSCRIPT

48. 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 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformat- ics), 2016, vol. 9677, pp. 37-48.

49. H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, Food Image Recognition Using Very Deep Convolutional Networks, in Proceedings of the 2nd International Workshop on Multime- dia Assisted Dietary Management - MADiMa 16, 2016, pp. 41-49.

50. Kawano Y, Yanai K ,“FoodCam-256: A large-scale real-time mobile food recognition system employing high-dimensional features and compression of classifier weights”., MM 14, 2014, 761-762.

51. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Work- shops, 2014, pp. 512-519.

52. L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst., vol. 106, no. 1, pp. 59-70, 2007.

53. G. M. Farinella, D. Allegra, and F. Stanco, A benchmark dataset to study the representation of food images, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 8927, pp. 584-599.

54. “Welcome to Python.org”, Python.org, 2017. [Online]. Available: https://www.python.org/. [Accessed: 24- Nov- 2017].
55. “scikit-learn Machine Learning in Python”, Scikit-learn.org, 2017. [On- line]. Available: http://scikit-learn.org/. [Accessed: 24- Nov- 2017].

56. S. Boseley, ”Global cost of obesity-related illness to hit $1.2tn a year from 2025”, the Guardian, 2017. [Online]. Available: https://www.theguardian.com/society/2017/oct/10/trea obesity-related-illness-will-cost-12tn-a-year-from-2025-experts-warn. [Accessed: 24- Nov- 2017].

57. C. Cusano, P. Napoletano and R. Schettini, ”Evaluating color texture de- scriptors under large variations of controlled lighting conditions”, Journal of the Optical Society of America A, vol. 33, no. 1, p. 17, 2015.

58. ”Pretrained CNNs - MatConvNet”, Vlfeat.org, 2018. [Online]. Available: http://www.vlfeat.org/matconvnet/pretrained/. [Accessed: 11- Feb- 2018].

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

59. Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Combining Deep Residual Network Features with Supervised Machine Learning Algorithms to Classify Diverse Food Image Datasets

Patrick McAllister[1], Huiru Zheng[1*], Raymond Bond[1], Anne Moorhead[2]

[1] *Ulster University, School of Computing; {mcallister-p2, *h.zheng, r.bond} @ulster.ac.uk*
[2] *Ulster University, School of Communication and Media; a.moorhead@ulster.ac.uk*

Abstract

Obesity is increasing worldwide and can cause many chronic conditions such as type-2 diabetes, heart disease, sleep apnea, and some cancers. Monitoring dietary intake through food logging is a key method to maintain a healthy lifestyle to prevent and manage obesity. Computer vision methods have been applied to food logging to automate image classification for monitoring dietary intake. In this work we applied pretrained ResNet-152 and GoogleNet convolutional neural networks (CNNs) to extract features from food image datasets; Food 5K, Food-11, RawFooT-DB, and Food-101. Deep features were extracted from CNNs and used to train machine learning classifiers including artificial neural network(ANN), support vector machine(SVM), Random Forest, fully connected Neural Networks, and Naive Bayes. Results show that using ResNet-152 deep features with SVM with RBF kernel can accurately detect food items with 99.4% accuracy using Food-5K food image dataset. Trained with ResNet-152 features, ANN can achieve 91.34%, 99.28% when applied to Food-11 and RawFooT-DB food image datasets respectively and SVM with RBF kernel can achieve 64.98% with Food-101 image dataset. From this research it is clear that using deep CNN features can be used efficiently for diverse food item image classification. The work presented in this research shows that pretrained ResNet-152 features provide sufficient generalisation power when applied to a range of food image classification tasks.

Keywords: obesity, food logging, deep learning, convolutional neural

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

networks, feature extraction

1. Introduction

Obesity is a global concern and is a serious health condition that can cause diseases such as heart disease, type-2 diabetes, and some cancers [1]. The increase of obesity has also been reported as a major burden on health care institutions through direct and indirect costs [56]. One of the major ways that obesity can be managed is through dietary management methods such as food logging and other methods [3]. Food logging is an activity in which the user document their energy intake to monitor their diet. Other methods may include the use of an exercise log book to document physical activities and the duration. Previously, users documented their intake using a food diary however many users now use smartphone applications to document their energy intake. The increase in smartphone usage has also led to the increase of well-being applications that are able to facilitate food logging. Many of these applications incorporate a simple diary entry, and/or connect to an online database/API to search for nutritional content for each of the users entries. Other novel methods include allowing the user to photograph the food items to determine calorie values. Using images has the potential to remove much of the complexity from traditional food logging to make it convenient for the user to document food intake to promote dietary management. Many studies have been completed in researching the use of computer vision methods to classify photographs of food to promote food logging [4-6]. This interactive approach to food logging using the camera within a smart-device may promote the use of food logging which is an important method to maintain weight loss. The remainder of this paper is structured as follows: Section 2 presents related work in how this problem has been tackled in previous research. Section 3 discusses the aim, objectives, and contributions of this work. Section 4 describes the methods used in this work and the use of Convolutional Neural Networks (CNNs) for feature extraction. Experiment results are presented in Section 5 followed by a discussion in Section

ACCEPTED MANUSCRIPT

6. Section 7 highlights study limitations and areas for future work.

2. Related Work

Food logging is a beneficial method to aid dietary management and recent novel methods have utilised meal photographs for food logging. A review [41] was completed to highlight a variety of computer vision methods that have been applied in food image recognition to promote dietary management. Key messages from this review are that there is a need for real food intake monitoring and one of the main challenges for diet monitoring using wearable sensors is practicability when used in a different environments and how automatic dietary monitoring is important to document nutritional intake habits to prevent conditions.

Food image recognition is a difficult task due to the amount of variation within food types. Food items in images are usually accompanied with other food items as well as other unrelated non-food items. The high variation of colour, shape, size, and texture in food items means that one method of image feature extraction and classification may not adapt to other foods and therefore a feature combination approach may be needed. Conventional ways to classify images utilise the use of hand-crafted feature extraction, e.g. global or local feature extraction using Speed-Up-Robust Features (SURF) [38] or local binary patterns (LBP) [39]. Feature engineering is used to determine what type of features and parameters are best used to successfully classify certain food types and categories and much work has been completed in this area. In [5] a bag-of-features model was proposed that used a combination of scale invariant feature transform (SIFT) features along with hue-saturation-value (HSV) colour features and a linear SVM to classify images into 11 categories with 78% accuracy. Other works also utilise a combination approach using SIFT and SPIN features and achieve high accuracy in classifying high level food groups (89% accuracy in classifying sandwiches and 91.7% in classifying chicken) using Pittsburgh

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Fast-Food Image Dataset (PFID). However, PFID dataset is an image dataset that was developed in a controlled laboratory environment, further works could be completed in applying this feature combination approach to similar image categories photographed in real-world environments. Other works use feature selection methods to determine optimal features [8] for food image classification. As well as using traditional feature extraction methods, CNN methods have become increasingly popular for image classification and this can be attributed to ImageNet Image Large-Scale Visual Recognition Challenge (ImageNet ILSVRC) as it allows users to compete against each other in achieving a classification accuracy and the winners in recent years have used convolutional neural networks (CNNs). Great emphasis has been placed on using CNNs for image classification and this is evident in a surge of recent research in this area relating to the fine-tuning CNN [11], deep feature extraction [12], and also training CNNs from scratch [11].

2.1. Detecting Food in Images Using CNN

CNN has been utilised for food image detection. This problem can be condensed down to a simple binary classification problem (food/non-food). The purpose of food image detection process is to first determine if food is present within an image or video. In regards to a food image recognition pipeline, this would be the first stage in food image recognition framework i.e. determining if the image contains food or not. In [13] GoogLeNet pretrained model was fine-tuned using Food-5K dataset. The training process in [13] utilised a subset of Food-5K data using 1000 iterations. The learning rate was changed to of 0.01 and the learning rate policy was polynomial. Results from [13] achieved 99.2% accuracy in determining food/non-food classes. Other research also utilised CNNs for food detection [14] and used 6-fold cross validation with different hyper-parameters to determine optimal settings and experiments achieved 93.8% in food/non-food detection.

ACCEPTED MANUSCRIPT

2.2. Predicting Food Type in Images Using CNN

Extensive research has been carried out in utilising CNN for food item recognition. The food item recognition process would take place after the food detection phase in which the actual food item is then predicted within the determined food image. In [15] CNNs were utilised to extract features from convolutional layers in order to determine if an image contains a food item and experiments achieved 70.13% for 61 class dataset and 94.01% for 7 class datasets, these experiments used AlexNet deep features with a SVM classifier applied to PFID dataset [15]. In [16] the aim of the work was to compare conventional feature extraction methods with CNN extraction methods utilising UEC Food 100 dataset. Results from [16] achieved 72.6% accuracy for top-1 accuracy and 92% for top-5 accuracy. Also in [14], as well as performing food/non-food experiments, food group classification was performed. A CNN was developed and was trained using extracted segmented patches of food items [14]. The food items used in this work were based around 7 food major types. The patches were then fed into a CNN using 4 convolutional layers with different variations of filter sizes and using 5 x 5 kernels to process the patches. Results in [14] achieved 73.70% accuracy using 6-fold cross validation. These studies confirm that CNN provide an efficient method for food image recognition to provide for accurate food logging to promote dietary management.

2.3. CNN Deep Feature Extraction Methods for Food Detection/Food Item Classification

Recent research has focused have used deep features extracted from pretrained CNN architectures to train machine learning classifiers for food image classification. Some research have opted for deep feature extraction opposing to fine-tuning pretrained CNN or training from scratch because less computational power and time is needed or small image datasets are used. Well-known CNN architectures (e.g. AlexNet, VGG-16, GoogleNet) for deep feature extraction have been developed in classifying images to automate food logging. This section discusses research that use deep feature extraction to detect food in images

ACCEPTED MANUSCRIPT

and classify food items in images for automated food logging. A comparative review was carried out on analysing the performance of a number of pretrained CNN architectures [43]. This review used VGG-S, Network in Network (NIN), and AlexNet for deep feature extraction to train food detection models. A food/non-food image dataset was collated and deep features were extracted from the models to train machine learning classifiers (one-class SVM classifier and binary classifier). Results showed that binary SVM classifiers trained with deep features achieved 84.95% for AlexNet, 92.47% for VGG-S, and Network In Network model achieving 90.82%. It is worth noting that UNICT-FD889 dataset used for deep feature extraction in [43] contains minimal noise as the images are focused on the food item, therefore this may contribute to high accuracy results. Further work could be completed in utilising a larger food image dataset consisting of images from different environments and also using different machine learning classifiers for further comparison.

Other research also explored the effect of training machine learning classifiers from different layers in pretrained AlexNet architecture [15]. Authors used AlexNet model to extract deep features from various layers deep in the architecture (FC6, FC7, and FC8 layers). The food image dataset used in [15] was PFID. Two experiments were presented in [15]; classifying high-level food catergories by organising PFID dataset into 7 category dataset and also classifying individual categories in PFID (61 classes). Results showed that the highest accuracy for the 61 class dataset was 70.13% using deep features extracted from layer FC6 in AlexNet. For the 7 class dataset, the highest accuracy achieved for deep features was 94.01% using layer from FC6. The contribution in [15] echoes the same findings in [43] suggesting that deep feature extraction provides high accuracies in classifying small grouped food image datasets (related food items) as well as datasets with specific different food types. Results also suggest that AlexNet deep features are able to efficiently generalise between high level food groups and also classify specific food groups with reasonable accuracy. However, more research needs to be completed in using deep features to classify food images in real world environments as PFID used in [15] was a laboratory

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

prepared dataset. As AlexNet is an early CNN architecture with a small amount of layers in comparison to more recent models, it was able to achieve reasonable accuracy in food item classification. AlexNet deep features from FC7 layer were able to achieve 57.87% using a standard linear SVM classifier classifying UEC-FOOD100 and 43.98% in classifying UEC-FOOD256 [45]. Fine-tuning AlexNet on a food image dataset and then performing deep feature extraction improved the accuracy to 67.57% in classifying UEC-FOOD256.

GoogleNet Inception CNN has also been used for deep feature extraction for food image classification [44]. Authors fine-tune a pretrained GoogleNet model using a food image dataset, and then deep feature extraction was used on another food image dataset. Experiments were completed in training a SVM using GoogLeNet deep features, in which the GoogLeNet model was fine-tuned using a food image dataset. Results showed that using deep features with SVM with PCA trained using fine-tuned GoogleNet features achieved 95.78% in classifying RagusaDB test set and 98.81% in classifying FCN test dataset which was an increase in accuracy comparison to other works using same datasets. The datasets used in [44] was small and more comparative research is needed in using a larger dataset of images photographed in different environments and real-world settings to fully evaluate the proposed approach [44].

In summary, previous research has showed that deep CNN features achieve high accuracies in determining food/non-food classification and classifying high level food groups[15,43,44,45]. It is also clear from the literature that deep CNN features from various CNN architectures at varying depths can easily distinguish between food/non-food and high level food groups. It has been suggested that deep features extracted from CNN should be an initial option in any visual recognition tasks [51], however in regards to food image classification, more work needs completed in exploring the use of next generation CNN architectures to extract deep features to train food classifiers, primarily for specific food item image classification photographed in real-world environments. This work compared the performance of using ResNet-152 and GoogleNet CNN deep features to classify a variety of food image datasets for food logging applications.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

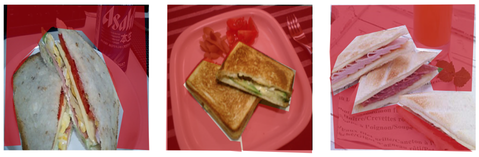


Figure 1: Example images of sandwiches from UEC FOOD 256 dataset highlighting noise in images.

3. Aim & Objectives

The aim of this work was to investigate the effectiveness of using deep feature extraction methods to classify variety of food image datasets to be used for dietary assessment. The work described in this paper seeks to answer the following research questions:

1. How efficient are deep residual network features for detecting foods in images and classifying food datasets using conventional machine learning algorithms?

2. How efficient are extracted GoogleNet deep features in predicting Food/Non-Food images and classifying images into high level food groups in comparison to fine-tuned GoogleNet model?

A series of experiments were completed that used the features extracted from CNNs and used them as input into conventional machine learning algorithms. To answer the research questions a number of objectives needed to be completed to achieve the aim of this work: (a) a number of oublic food image datasets needed to be selected, (b) several pre-trained CNNs needed to be identified from the literature for deep feature extraction, (c) supervised machine learning algorithms needed to be identified to classify the images using the extracted deep activations; and (d) statistical analysis is then applied to the results to

ACCEPTED MANUSCRIPT

evaluate the methods used. The next section will discuss in detail the methods used in this work.

4. Methodology

4.1. Food Image Datasets

In this work we identified publicly available food image datasets to use for the experiments to determine efficiency of using pretrained CNNs to extract deep features for image classification. The following image datasets were used in this work (Table 1):

1. Food-5K
2. Food-11
3. RawFooT-DB
4. Food-101
5. UNICT-FD889

Table 1: Table showing name, number of categories, images per category, as well as how the image datasets were developed of each food image dataset.

Dataset	**Catergories**	**Images Per Catergory**	**Image Preparation**
Food-5K [13]	2	2500 (training set) 500 (val & eval sets)	Real world
Food-11 [13]	11	Unbalanced	Real world
RawFooT-DB [20]	68	368 each in training/test set	Controlled
Food-101 [17]	101	1000	Real world

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4.2. Food-5K

Food-5K dataset consisted of 2 categories; food and non-food, training is balanced and contains 2500 images of each category [13]. The dataset also contains a validation and evaluation set and each category contains 500 images each per dataset. The authors developed this dataset to measure the performance of using GoogLeNet pretained CNN for classification. Food-5K was developed by selecting images from already public available datasets e.g. Food-101 [17], UEC-FOOD100 [18] and UEC-FOOD256 [19]. The authors described this dataset as being varied as they selected foods that cover a wide variety of different food dishes. The images also contain some noise and multiple food items may be contained in an image. The non-food images consisted of images that do not contain food items (objects or humans). Food-5K was used to find out how ResNet-152 deep features perform in detecting food items in images, which can be argued is an important first step in food image classifcation for food logging. The authors developed the non-food image dataset from using other publicly available datasets e.g. Caltech101, Caltech256, Emotion6, and Images of Groups of People.

4.3. Food-11

Food-11 is a dataset that comprises of 11 major food groups [13]. The 11 categories are diary, bread, egg, dessert, meat, fried food, pasta, seafood, rice, vegetables/fruit, and soup. Food-11 dataset was also created using images from Food-101, UEC-FOOD-100, and UEC-FOOD-256. The authors of Food-11 stated that the images selected cover a wide range of food types in order to train a strong classifier that had the ability to classify different varieties of foods. Many of the images contained in Food-11 were taken in real world environments, therefore the images contain high colour variation and some noise (unrelated food items) may be present. The developers of this dataset have divided the dataset into training, validation, and evaluation similar to Food-5K. Food-11 was used to explore the performance of ResNet-152 deep features in categorising food images using Food-11.

ACCEPTED MANUSCRIPT

4.4. RawFooT-DB

RawFooT-DB [20,42] food image dataset was developed to research the use of computer vision methods to classify food image textures under different lighting conditions. Each image in RawFooT-DB is unique in regards to the light direction, light intensity, and colour illumination and food image textures are isolated with no noise or other food items present. The dataset contains 68 classes with wide variety of food types ranging from fish, meat, fruit, and cereals. RawFooT-DB dataset contains tiles from the images in the RawFooT-DB. Each image is divided into 16 tiles, 8 tiles are for training and the remaining 8 for testing. Each class contains 368 images (tiles) which represent 8 tile texture samples under 46 different lighting conditions. In this research, we explored the use of ResNet deep feature features to train machine learning classifiers. RawFooT-DB was used to explore how ResNet-152 deep features perform in generalising food texture between class variance. Previous research divided RawFooT-DB into different lighting condition subsets [20, 42], in this work we explored the performance of using ResNet-152 deep features across multiple lighting conditions and each food class in RawFooT-DB contains multiple food texture patches across different lighting conditions.

4.5. Food-101

Food-101 consists of 101 food categories and each category contains 1000 images [17]. The Food-101 dataset have been described as challenging as much of the images in the dataset contain noise and the images were collated from Foodspotting, which is a social media website that allows users to upload food images. This means that images used are from a real-world setting i.e. restaurant or at home and not in a lab environment. Food-101 allows us to research how ResNet-152 deep features performs in classifying food items with similar food dishes in varying real world environments. Authors of Food-101 specify dedicated training and testing splits with testing splits containing images that are 'cleaned' of noise, in this work we also use 75:25 training/testing partitions, however data was shuffled before partition for preliminary analysis to determine

ACCEPTED MANUSCRIPT

how ResNet-152 features perform in classifying images with noise and intense colour and food variation. Figure 2 illustrates an example of the images in the datasets.

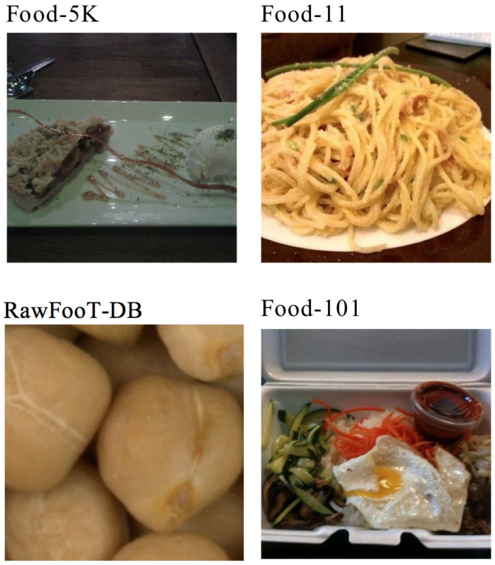


Figure 2: Example of images from 4 food image datasets used in this work.

4.6. Datasets for Further Evaluation of Food/Non-Food Detection Models

Due to the small size of Food-5K, two other datasets have been used to evaluate our trained food/non-food models; UNICT-FD889, which is a food image dataset, and Caltech, which is a non-food image dataset. Deep features were extracted from UNICT-FD889 and Caltech and classified by models that achieved the best performance in classifying Food-5K datasets.

UNICT-FD889

UNICT-FD889 (Figure 3) was used to evaluate food/non-food models trained using Food-5K [53]. UNICT-FD889 contains 889 distinct food dishes to study food representation and the images are photographed in real world environments which means that much of the images may contain high food variance, however the images in UNICT-FD889 contain images that are focused on the food item with little noise.

ACCEPTED MANUSCRIPT

Figure 3: Example of images contained in UNICT-FD889 dataset.

Caltech-101

Caltech-101 dataset (Figure 4) was also used for evaluating food/non-food classification models. Caltech-101 contains 101 image categories and each contains between 50-800 images. The categories are non-food based and contain images relating to animals and objects and each image is around 300x200 pixels in size [52].

Figure 4: Example of images contained in Caltech-101 dataset.

4.7. Overview of Convolutional Neural Networks

The use of pretrained CNNs gives great potential for applying them to a variety of problem areas. Convolution is used to describe the type of neural network as the input image is broken down into smaller overlapping shapes in order to determine certain patterns in the image. These overlapping segments are called filters. The patterns detected, by each overlapping shape in the filter, may consist of a colour contrast or certain interest points such as edges. The overlapping shapes look for the same pattern on the image. The overlapping tiles are effectively used as input for a small neural network. This is done for each tile in the image. Each network in the filter hold the same weights to determine interest points in each tile. The output of this process is an array

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

which each section corresponds to the network that describes patterns in each tile. A down-sampling process is then triggered after the convolution stage, this is typically completed using max pooling where the representation divided into non-overlapping rectangles. Within each region the maximum is retained. This process can be repeated a number of times to create deeper and more detailed representations. Fully connected layers are also present with a CNN architecture and is connected to activations from the layer previous. The fully connected layer takes the input from previous layers and uses this for classification using a soft-max function. Backpropagation is typically used to train the CNN in which the forward propagation is used to determine the error and gradient descent is then used to update the weights and parameters based on this error. This is repeated in order to train the CNN using a training dataset [21,22].

4.8. Image Preprocessing for Feature Extraction

The pretrained CNNs used in this work were trained specifically with requirements placed on the input images. Therefore, in order to extract deep feature representations of these images using these CNNs, it was important to ensure that the images meet the same requirements. The first requirement was to ensure that the images were resized to a specific height and width configured in the image input layer of the pretrained CNN. The images are also normalised and this is achieved by subtracting the mean of the image. The mean is removed from the input image and also the image intensities are normalised within a [0,255] region, as defined in [23].

4.9. Deep Feature Extraction

In this work we used 2 pretrained CNNs as deep feature extractors. The advantage of using a pretrained CNN to extract deep image features, as opposed to training a new CNN, are: (1) less computational power is needed as we are allowing the CNN to process each image only once to extract deep feature representations; (2) less data is needed in order to achieve high accuracy results as layers deep in the CNN architecture contain activations that can be used for

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

deep feature representations.

CNNs have been trained to specifically determine and highlight key features in an image and pretrained CNNs allow images to be inserted and layers produce a response or activation to the image. These 'activations' or deep features as they will be called in this work, can be extracted in the form of a feature vector [23,24]. The authors that created datasets Food-5K and Food-11 fine-tuned a GoogLeNet model, therefore for performance comparison, we adopted a different approach of using GoogLeNet, not for fine-tuning but for deep feature extraction and to use these deep features to train machine learning classifiers. As stated, the 2 CNNs we have chosen achieved high accuracy results when applied to ILSRVC ImageNet dataset.

Comparing this feature extraction process to training a CNN from scratch, in which mini-batches of image data are iteratively passed through different layers (i.e. convolutional and sub-sampling layers) using back-propagation to implement stochastic gradient descent to train the network, the method of deep feature extraction requires less computational power. Deep feature extraction can also be implemented on a CPU as only one pass is completed through the training data to extract the deep features. It is also worth noting that a large amount of time needs to be dedicated to train a CNN from scratch. For many researchers this is not possible, therefore pretrained CNNs offer a convenient way to experiment with deep learning algorithms by allowing for deep feature extraction, classification, and also transfer learning.

The datasets used in this work are small in comparison to the datasets needed to train a CNN from scratch such as ILSRVC dataset which contains over 14 million images [59]. Figure 5 describes the pipeline used in this work where by images are processed to extract deep features to be used for classification.

ACCEPTED MANUSCRIPT

4.9.1. Layer Selection

To extract features from pretrained CNN, a layer needs to be selected for each model. During the training of CNN models, the output from convolutional layers and the pooling layers depict high level representations of images. In this study we extracted deep feature maps immediately after the last pooling layer of each CNN to determine if these feature representations are able to accurately generalise between different food classes in food image dataset. The layer names used to extract deep features from CNN architecture are used to distinguish between different layers in the pretrained CNN models. Table 2 lists the size of each pretrained CNN model and the chosen layer for deep feature extraction.

Table 2: Table showing pretrained CNN used as deep feature extractors in this work. The table lists the name of the CNN, the amount of layers present, the dataset used to train the CNN, and layer used in this work.

CNN	Layers	Trained	Layer
ResNet-152	152	ImageNet ILSVRC	pool5
GoogLeNet	22	ImageNet ILSVRC	cls3_pool

4.10. Pretrained Models using MatConvNet Package

MatConvNet is a popular Matlab library that allows for the training of state-of-the-art CNNs or to apply pretrained CNNs for deep feature extraction to be used for image classification [23,24]. In this work, MatConvNet was used to utilise 2 pretrained CNNs for deep feature extraction both trained on ILSVRC ImageNet dataset. MatConvNet packages allow for the fine-tuning of pretrained CNN [24]. In this work ResNet-152 and GoogLeNet were chosen to extract deep features to train classification models, the reason ResNet-152 was used was that it has achieved the lowest top-1 error of 23% using ILSVRC 2012 validation dataset in the MatConvNet package. GoogLeNet is another popular model available on MatConvNet package and was used for deep feature extraction in

ACCEPTED MANUSCRIPT

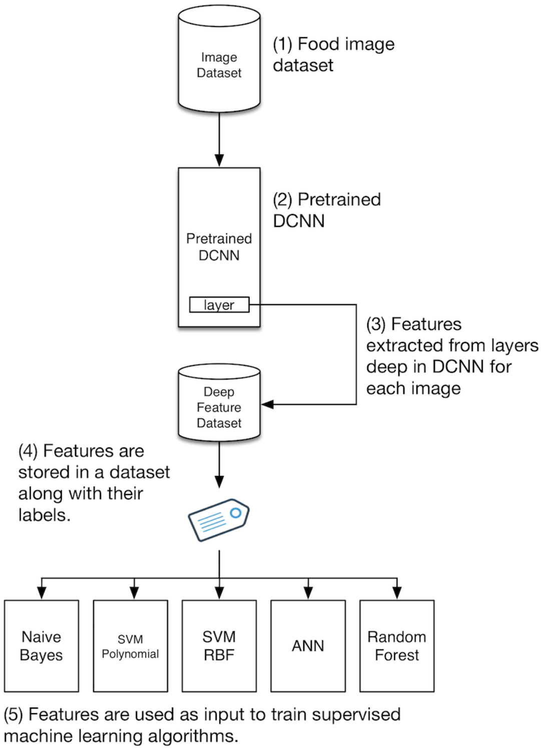


Image Dataset
(1) Food image dataset
Pretrained DCNN
layer
(2) Pretrained DCNN
(3) Features extracted from layers deep in DCNN for each image
Deep Feature Dataset
(4) Features are stored in a dataset along with their labels.
Naive Bayes
SVM Polynomial
SVM RBF
ANN
Random Forest
(5) Features are used as input to train supervised machine learning algorithms.

Figure 5: Diagram describing the pipeline of deep feature ex- traction. (1) Food image datasets are used as input into (2) (pretrained CNN). (3)A layer deep in the architecture is specified and the image is processed by the CNN and the output (of the specified layer) is a generic image feature vector. (4) These generic image feature vectors can be collated to form a feature dataset and each feature vector generated by the CNN layer is labelled in accordance to the category from where the image taken from. (5) The generic image feature dataset can then be used as input to a range of conventional machine learning algorithm.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

this work for performance comparison with the fine-tuned GoogleNet model trained in [13].

4.11. ResNet-152 CNN

ResNet-152 is a deep residual pretrained CNN [25]. At the time of development, the authors of this CNN have described it as the deepest network ever presented on ImageNet (2015) and is based on utilising extremely deep nets with a depth of up to 152 layers. A residual learning framework which allows training of networks easier to converge and promote increased accuracy. The main advantages that residual networks contribute is the acceleration of speed in training networks, the effect of the vanishing gradient problem is reduced, and increasing the depth of the network which results in less parameters. ResNet-152 is made up of residual connections that allow important information to be transferred between layers. Residual connections allow a gradient to pass backwards directly through layers without losing vital information, in a regular CNN, the gradient must always pass through an activation layer. This can cause the gradient to diminish, to circumvent this problem, connections within a CNN are appended with a shortcut that allows gradients to pass through thus decreasing the effects of vanishing gradient (information loss). Experiments using residual connects (ResNet-152) have reported increased accuracy and lower training times, in comparison to other state of the arts [25]. The authors of ResNet-152 compare their work with other established CNNs and state that this residual deep net is 8x deeper than VGG nets [26]. We used ResNet-152 pretrained CNN with the image datasets mentioned in this work for feature extraction. We selected pool5 layer deep in the ResNet-152 architecture and for each image an extracted a feature vector of 2048 was computed.

4.12. GoogleNet - Inception

GoogLeNet was used for deep feature extraction combined with the same supervised machine learning models. In [22] a deep convolutional network was

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

proposed that is able to achieve state of the art classification and object detection accuracy by training the network using ImageNet dataset for Large Scale Visual Recognition Challenge 2014. The motivation for GoogLeNet was that larger CNNs may encounter the problem of overfitting as there is a large number of parameters used in the network. GoogLeNets main contribution is the introduction of Inception modules that utilises the concept of using approximation of sparse structure with repeated dense components. Dimensionality reduction is used in order to ensure computational complexity is kept to a minimum. Multiple convolutional filters are used with different sizes to ensure that there is sufficient coverage of information clusters. Before more computational expensive convolutions (3x3, 5x5) a convolutional after the previous layer for data reduction. The results of GoogLeNet incorporating these inception modules achieved 6.67% top-5 error percentage in classification performance in ILSVRC Classification Challenge 2014. In this work, we extracted the deep activations using the fully connected layer cls3_pool which has a 1024 vector dimension and is located after the last pooling layer in GoogLeNet [22].

4.13. Metrics for Performance Measurement

Several metrics were used to assess the performance of the trained models. The metrics that were selected to assess each model were percentage, recall, F1 score, Kappa, and Area Under the Receiver Operating Characteristic curve (AUC). The output of each model can be presented using a confusion matrix. A confusion matrix is a table that is able to summarise the prediction outcome of a model by classifying instances as positive (P) instances or negative (N) instances. Confusion matrix can further provide greater insight into prediction outcomes by classifying predicted instances as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Visually, the performance of a confusion matrix can be quickly assessed by inspecting the diagonal line of the confusion matrix, the stronger instances that are present in this diagonal line signifies better performance. The metrics used to assess the experiments can be derived from the confusion matrix such as recall (sensitivity), Ac, and

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

F1 score. Recall can be described as metric that describes how many instances are classified correctly. The F1 score is a weighted average using precision and recall and is measured between 0 (worst) and 1 (best). For Food-5K the AUC values were also computed for each experiment due to being a binary classifier and Cohen's kappa was calculated for Food-11, RawFooT-DB, and Food-101. Cohen's kappa is a metric that is used to measure the inter-rater agreement between two label sets in a classification problem, we use Cohen's Kappa along with other metrics to describe experiment results [27].

4.14. Training, Validation, and Evaluation Data Partitions

To evaluate the performance of our trained models, validation and evaluation datasets were extracted and used from Food-5K, and Food-11. For RawFooT-DB, an evaluation dataset was used supplied by the authors [20]. For Food-5K, Food-11, and RawFooT-DB, the authors already partitioned the datasets into evaluation and validation sets (Table 3) and in this work we used the same data splits to train and test our models. For Food-101, we split the data into 75:25 for training and testing. Authors of Food-101 provide training and testing splits with testing images cleaned of noise, however in this work we randomly shuffled images for training and testing partitions to test how ResNet-152 performs in classifying food images with noise and high food variance. This would give an indication of how deep features would perform in classifying difficult datasets such as Food-101. Table 3 is a summary of the data partitions used in this work for each food image dataset and the names for each partition follows the author's naming convention. Several metrics were computed during the experiment stage e.g. kappa statistic, F1 score, recall, average ROC, and accuracy to measure the performance of each trained model. Food-5K and Food-11 datasets each contained training, validation, and evaluation images. Training images were used for feature extraction to train machine learning classifiers. Validation images were used to determine if hyper-parameters used yield adequate results and evaluation dataset was to fully evaluate overall trained model. For RawFooT-DB, authors developed training and testing datasets by taking each image and

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

dividing it into 16 tiles, 8 tiles are for training and the remaining 8 for testing. Each class contains 368 images (tiles) which represent 8 tile texture samples under 46 different lighting conditions. The testing dataset was used to verify if the trained model able to generalise between food texture classes. Food-101 dataset was randomly partitioned; 75% for training and 25% for testing. Testing partition was used to verify trained Food-101 classifiers. UNICT-FD889 and Caltech-101 testing datasets were used to further evaluate food/non-food classification models.

Table 3: Table showing testing methods used for each food image dataset. * denotes dataset splits supplied by dataset authors.

Dataset	**Dataset Partition**
Food-5K	Training, validation & evaluation*
Food-11	Training, validation & evaluation*
RawFooT-DB	Training & testing*
Food-101	75:25 training & testing
UNICT-FD889	Testing
Caltech-101	Testing

4.15. UNICT-FD889 & Caltech-101 Food/ Non-Food Dataset

As well as using the validation and evaluation datasets supplied with Food-5K, further evaluation was completed with UNICT-FD889 dataset and Caltech-101 dataset in detecting food images. UNICT-FD889 is a food dataset containing images from a range of food types and Caltech-101 is a non-food image dataset, UNICT-Caltech. These 2 datasets were combined to create a new food/non-food dataset called UNICT-FD889 to evaluate our food detection models. Deep features were extracted from the new food/non-food dataset. Further evaluation was completed because Food-5K evaluation and validation

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

datasets are small with only 500 images in each category for each dataset. Using another larger dataset for evaluation can give a stronger performance indication of our models in classifying a large variety of food and non-food images.

4.16. Platform

In order to train the machine learning algorithms, Weka 3.8.1 [28] platform was used. Weka is a software application that contains various machine learning algorithms written in Java and the application was developed at University of Waikato, New Zealand. The application can be used for different tasks such as clustering, classification, visualisation, feature selection, and preprocessing and is very popular within universities for its ease of use. It is also popular because of the amount of algorithms available. The main reason that Weka 3.8.1 was used in this work was the detailed evaluation results output computed, which are collated into a window after evaluation has finished. Another major advantage of using Weka is the evaluation process in that a range of detailed metrics are computed for each class to describe the performance of the model. A confusion matrix can be computed to determine the performance of individual classes for the trained model using K-fold class validation or a dedicated validation dataset. The amount of machine learning algorithms that are available is a factor in using Weka as well the easy to use graphical user interface (GUI). In this work, Weka 3.8.1 was used with the extracted features from image datasets for classification, analysis, and evaluation [28].

4.16.1. WekaPython Plugin & Scikit-Learn

WekaPython plugin was used with Weka 3.8.1 that allows the training of scikit-learn [29,55] machine learning classifiers. The wekaPython package relies on Python version 2.7 or higher being installed on the user's system and uses a range of Python packages to function correctly such as pandas, numpy, scikit-learn, and matplotlib. In this work, the wekaPython was used to train and evaluate the deep features extracted from the pretrained CNNs. Weka was used to train an ANN for experiments with Food-101. Due to its flexibility for

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

working with larger datasets, Python v2.7.10 with scikit-learn library was also used to train the other machine learning classifiers for the Food-101 dataset [30]. The following machine learning algorithms were used in this work [29,54]:

1. Gaussian Naive Bayes (wekaPython scikit-learn)
2. Support Vector Machines (SVM) (wekaPython scikit-learn)
3. Artificial Neural Network (ANN)
4. Random Forest Classifier (wekaPython scikit-learn)

For Food-101 food image dataset, datasets were manually split 75:25 and the follow parameters were used to split and shuffle the dataset to train and test each machine learning classifier;

1. Gaussian Naive Bayes - random_state 1
2. Support Vector Machines - random_state 1
3. Artificial Neural Network - random_seed 1
4. Random Forest Classifier - random_state 1

4.16.2. Naive Bayes

Naive Bayes is a popular machine learning algorithms known for their efficiency and minimal processing. They can be described as a set of simple probabilistic classifiers derived from Bayes Theorem. The term naive is used to describe the algorithm because it assumes that attributes are independent of the associated class. Bayes rule is enforced to compute the probability of a class based upon the values in the vector. Bayes rule of conditional probability states that if you have a hypothesis H and the evidence (feature attributes) is connected to that hypothesis [31]. Naive Bayes assumes independence and the algorithm works efficiently and can outperform the most sophisticated machine learning algorithms on certain datasets. Naive Bayes can be described as a simplistic approach to using learning probabilistic knowledge for classification. However, the present of redundant data can affect the performance and the introduction dependent attributes also diminish the performance of classifier.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

In this work, a Gaussian naive bayes classifier was trained using the extracted CNN deep features. A Gaussian naive bayes classifier is used when continuous values are present by assuming a normal distribution in the dataset as the mean and standard deviation is computed for each class.

4.16.3. Support Vector Machines (SVM)

SVMs are able to implement the use of non-linear boundaries by using kernels (e.g. RBF, Polynomial) to transform feature representation into a higher dimensional space to predict multiple classes. In classification problems, the use of SVM have performed well in generalising on a variety of classification problems such as food classification, face detection, and object detection [32,33]. In some problems the training data in a problem may become inseparable meaning that there is not a clear boundary definition, SVMs are able to enforce nonlinear boundaries in transformed feature spaces [35]. In regards to a linear SVM, a linear hyperplane is computed and considered optimal if a line is at a furthest distance from class data points (largest minimum distance) [35]. However, in some instances the training data may not be linearly separable, therefore SVM employ the use of kernels to determine optimal hyperplanes. Kernels can be used in order to fit linear models in a non-linear setting, mapping is used to transform how the features are represented into a higher dimensional space. In this work, we train 2 C-SVM models using Polynomial kernel and Radial Basis Function (RBF). C-SVM uses a C regularisation parameter that implements a weight penalty for misclassifications to improve the accuracy of the model.

4.16.4. Artificial Neural Network (ANN)

An ANN or feed-forward neural network was also used in this work and ANN can comprise of a number of layers. Each layer contains a number of nodes that are called neurons. The basic ANN architecture is made of three layers; input layer, hidden layer, and output layer and because of the amount of rich information/features that can be learned using a ANN, it can be applied to problems that are of an non-linear nature. The basic function of a ANN is

ACCEPTED MANUSCRIPT

the ability to map features data into a set of outputs. Each neuron computes its input by using a weight that represents the strength between nodes. An activation function is then applied, there are a number of activation functions that are available i.e. sigmoid function, linear, or gaussian. Once the activation function is applied, a single value is returned. Back propagation is used to train the ANN, the predicted output is compared to the expected output which is reflected in the cost function and the weights are altered. ANN training can be customised to suit the nature of the input dataset and problem, parameters such as training time (epochs), learning rate, and momentum can be configured. In this work, ANNs were trained for each dataset using a Weka plug-in [30] with the following parameters listed in Table 4. The learning rate was set to adaptive unless otherwise stated in the experiments. The adaptive learning rate function uses a number of base learning rates on the training data to determine the most suitable by comparing the cost function of each. The Weka plugin uses dropout regularisation to prevent overfitting and Rectified Linear Units as the activation functions [30,36].

Table 4: Hyper-parameters used for each ANN.

ANN	**Parameters**
Number of iterations	1000 (max)
Num of layers	1
Neurons per layer	100
Learning rate	Adaptive*
Learning momentum	0.2
Weight penalty	0.00000001 (default)
Hidden Layers drop out rate	0.5
Input layer drop out rate	0.2
Activation function	ReLu
Convergence threshold	0.2
Batch	100

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4.16.5. Random Forest

Random Forests (RF) was developed by Leo Brieman and Adele Culter [37] and is a classification algorithm that utilises a number of decision trees using feature subsamples and bootstrapped examples. The purpose of RF was to be easy to use by offering little preprocessing requirements and using a voting system for final classification using a collection of decision trees. This method is directly related to the bagging technique as the goal of the bagging technique is to develop a model with low variance and to average noise in the dataset. RF is able to take subsets of the input data comprised of random values with each instance labelled with its class. For each subset created a decision tree is created, as depicted in (1).

$$D = \begin{bmatrix} i_{\mathrm{a}1} & i_{\mathrm{b}1} & i_{\mathrm{c}1} & c_1 \\ i_{\mathrm{a}2} & i_{\mathrm{b}2} & i_{\mathrm{c}2} & c_2 \\ i_{\mathrm{a}3} & i_{\mathrm{b}3} & i_{\mathrm{c}3} & c_3 \end{bmatrix} \tag{1}$$

$$\begin{aligned} D_1 &= \begin{bmatrix} i_{\mathrm{a}1} & i_{\mathrm{b}1} & i_{\mathrm{c}1} & c_1 \end{bmatrix} \\ D_2 &= \begin{bmatrix} i_{\mathrm{a}2} & i_{\mathrm{b}2} & i_{\mathrm{c}2} & c_2 \end{bmatrix} \\ D_3 &= \begin{bmatrix} i_{\mathrm{a}3} & i_{\mathrm{b}3} & i_{\mathrm{c}3} & c_3 \end{bmatrix} \end{aligned} \tag{2}$$

In (2), each decision tree D is trained using the subset training data and a classification for each instance is calculated. A majority voting rule is then used to decide on the final classification of the instance. Random Forest algorithm is efficient in that it is able to analyse large databases and is able to estimate missing data to help maintain accuracy [37]. In this work a scikit-learn Random Forest classifier was used with wekaPython and Table 5 lists the parameters used for this model.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 5: Table showing hyper-parameters used for Weka Random Forest classifier. Hyper-parameters used for this classifier are default.

Random Forest	Parameters
Criterion	entropy
Number of estimators	50
Random State	none
Depth of tree	None
Minimum number of samples split	2
Minimum number of samples for leaf node	1
Number of features for best split	auto
Bootstrap	True
Max leaf nodes	None
Random State Instance	None
Max depth	None
Minimum num of leaf samples	1

5. Experimental Results

5.1. Food /Non-Food Classification Results

5.1.1. Food-5K

This section lists the results of our experiments using the food image datasets. Tables 6 and 8 list the detailed results of Food-5K. Accuracy, recall, F1 score, and ROC values were used to measure the performance of each the classification models for both validation and evaluation datasets. Initial results show that deep features combined with machine learning classifiers achieved high accuracy results when distinguishing between food and non-food images. The use of SVM with RBF kernel achieved the highest accuracy with 99.4% using ResNet-152 for deep feature extraction with validation dataset and 98.8% with evaluation dataset. Table 7 and 9 also lists the confusion matrices of using SVM-RBF with ResNet-152 to detect food images in validation dataset and ANN with

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

ResNet-152 features to detect food images in evaluation dataset. GoogLeNet deep features achieved marginally lower accuracy results, however for the evaluation dataset, GoogLeNet deep features with ANN achieved the same accuracy result as SVM-RBF and Random Forests classifier with ResNet-152 features with 98.8%. In regards to using SVM classifiers in Food-5K, the use of the RBF kernel achieved marginally higher accuracies compared to the polynomial kernel and Gaussian naive bayes achieving the lowest accuracy results in both testing datasets with both deep feature types.

Table 6: Classification results using ResNet-152 and GoogLeNet to extract deep activations (extracted from Food-5K) with supervised learning algorithms. Figures in bold represent highest accuracy result.

Food-5K - Validation								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3_pool**			
	Acc (%)	**Recall**	**F1**	**ROC**	**Acc (%)**	**Recall**	**F1**	**ROC**
NB	98.7	0.99	0.99	0.99	97.5	0.98	0.98	0.99
SVM (RBF)	**99.4**	0.99	0.99	0.99	98.5	0.99	0.99	0.99
SVM (Poly)	99	0.99	0.99	0.99	98.5	0.99	0.99	0.99
ANN	99.2	0.99	0.99	1	99	0.99	0.99	0.99
RF	98.9	0.99	0.99	1	98.6	0.99	0.99	0.99

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 7: Confusion matrix showing results of highest accuracy results achieved using ResNet-152 features classifying validation dataset of Food-5K using a SVM with RBF kernel.

Confusion Matrix using SVM-RBF with ResNet-152 Validation Dataset Features

		Predicted Labels	
		Food	Non-Food
True	Food	498	2
	Non-Food	4	496

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 8: Classification results using ResNet-152 and GoogLeNet to extract deep activations (extracted from Food-5K) with supervised learning classifiers using evaluation dataset.

Food-5K - Evaluation								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3_pool**			
	Acc (%)	**Recall**	**F1**	**ROC**	**Acc (%)**	**Recall**	**F1**	**ROC**
NB	97.3	0.97	0.97	0.98	96	0.96	0.96	0.98
SVM (RBF)	98.8	0.99	0.99	0.99	98.3	0.98	0.98	0.98
SVM (Poly)	98.3	0.98	0.98	0.98	98.2	0.98	0.98	0.99
ANN	**98.8**	0.99	0.99	0.99	**98.8**	0.99	0.99	0.99
RF	98.8	0.99	0.99	0.99	98.5	0.99	0.99	0.99

Table 9: Confusion matrix showing results of highest accuracy results achieved using ResNet-152 features classifying evaluation dataset of Food-5K using ANN.

Confusion Matrix using ANN with ResNet-152 Evaluation Dataset Features

		Predicted Labels	
		Food	Non-Food
True	Food	493	7
	Non-Food	5	495

To further test our models, experiments were conducted that tested food/non-food trained models on the Food-11 dataset as what was completed in [13] for more detailed comparison. Food-11 dataset contains 16,643 images and they are all classed as food images, GoogleNet and ResNet-152 deep features were used to extract deep features from Food-11 and used with SVM-RBF and ANN models to classify them to detect food in the images. Table 10 is a breakdown

ACCEPTED MANUSCRIPT

of the results using our methods to classify Food-11 dataset.

Table 10: Results comparison of classifying Food-11 with our Food/Non-Food classification models..

Method	Num of Food Images Detected	Accuracy
ResNet-152 + ANN (Food-11)	16, 208	97.39%
ResNet-152 + SVM-RBF (Food-11)	16,176	97.19%
GoogleNet + ANN (Food-11)	16,171	97.16%
GoogleNet + SVM-RBF (Food-11)	15,646	94.01%
ResNet-152+ SVM-RBF (UNICT-Caltech)	12,409	97.50%
ResNet-152+ ANN (UNICT-Caltech)	12,283	96.51%

5.1.2. UNICT-FD889 & Caltech

Table 10 list the results of using SVM-RBF and ANN trained with Food-5K training ResNet-152 deep features for classifying UNICT-Caltech, which combines images in UNICT-FD889 and Caltech-101 to make a food/non-food dataset. UNICT-Caltech dataset is a larger dataset and using this dataset with our trained models allows us to get a better indication how ResNet-152 features perform in detecting food in images.

5.2. Food Item Classification Results

5.2.1. Food-11

Results show that using ResNet-152 and GoogleNet deep features are able to achieve high accuracies when classifying across major food groups. Results are presented in Tables 11 and 12. The maximum accuracy achieved was using ANN for both ResNet-152 and GoogleNet features achieving 91.34% and 86.44% respectively with evaluation dataset. For ResNet-152 features an F-measure of 0.91 was achieved and 0.86 with GoogleNet features using ANN. For the ANN

ACCEPTED MANUSCRIPT

trained using ResNet-152 features, the base learning rate was set to auto-detect which allows the ANN Weka plugin to initially test various learning rates to determine the lowest cost function. Initial tests revealed that 1.0 learning rate achieved the lowest cost function and the ANN used that to learning rate to initially begin the training. The learning rate decreased over the course of the training if the network cost function didn't improve after 10 mini-batch iterations. The network converged after 204 iterations ending with a learning rate of 0.01. Further analysis revealed the SVM models trained with RBF and Polynomial kernel using ResNet-152 features achieved 89.99% and 88.86% accuracy respectively and 85.36% and 86.05% using GoogleNet features using evaluation dataset. Figure 6 shows the confusion matrix of using an ANN trained with ResNet-152 features to classify the evaluation dataset. Figure 7 is an example of different types of food categories that were misclassified as shown in the confusion matrix in Figure 6.

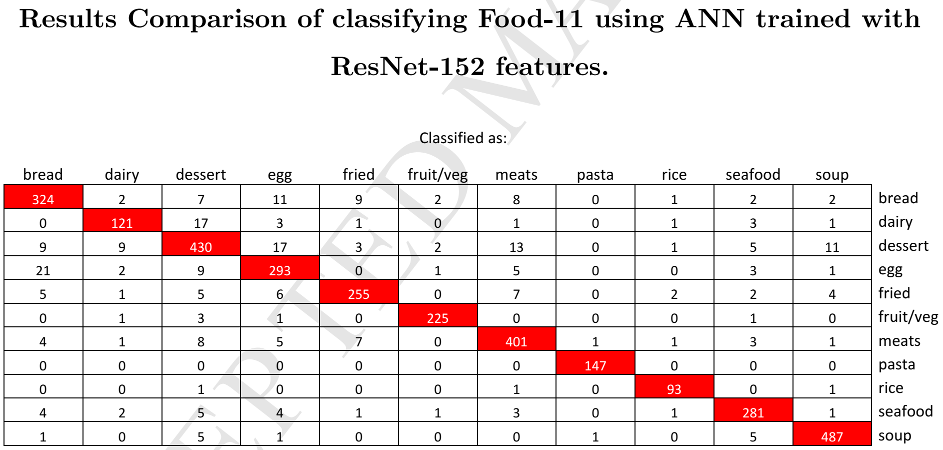
Table 11: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from Food-11) with supervised learning classifiers.

Food-11 - Validation Dataset								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3_pool**			
	Acc (%)	**Recall**	**F1**	**Kappa**	**Acc (%)**	**Recall**	**F1**	**Kappa**
GNB	73.03	0.73	0.73	0.70	67.49	0.68	0.68	0.64
SVM (RBF)	88.11	0.88	0.88	0.87	82.36	0.82	0.82	0.80
SVM (Poly)	86.65	0.87	0.87	0.85	83.70	0.84	0.84	0.82
ANN	**89.18**	0.89	0.89	0.88	84.11	0.84	0.84	0.82
RF	78.43	0.78	0.78	0.76	75.48	0.76	0.75	0.72

ACCEPTED MANUSCRIPT

Table 12: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from Food-11) with supervised learning algorithms.

Food-11 - Evaluation Dataset								
Model	**ResNet-152 - pool5**				**GoogLeNet - cls3_pool**			
	Acc (%)	**Recall**	**F1**	**Kappa**	**Acc (%)**	**Recall**	**F1**	**Kappa**
GNB	75.38	0.75	0.76	0.72	69.73	0.70	0.70	0.66
SVM (RBF)	89.99	0.90	0.90	0.89	85.36	0.85	0.85	0.84
SVM (Poly)	88.86	0.89	0.89	0.87	86.05	0.86	0.86	0.84
ANN	**91.34**	0.91	0.91	0.90	86.44	0.86	0.86	0.85
RF	80.40	0.80	0.80	0.78	78.24	0.78	0.78	0.75



Results Comparison of classifying Food-11 using ANN trained with ResNet-152 features.

Classified as:

bread	dairy	dessert	egg	fried	fruit/veg	meats	pasta	rice	seafood	soup	
324	2	7	11	9	2	8	0	1	2	2	bread
0	121	17	3	1	0	1	0	1	3	1	dairy
9	9	430	17	3	2	13	0	1	5	11	dessert
21	2	9	293	0	1	5	0	0	3	1	egg
5	1	5	6	255	0	7	0	2	2	4	fried
0	1	3	1	0	225	0	0	0	1	0	fruit/veg
4	1	8	5	7	0	401	1	1	3	1	meats
0	0	0	0	0	0	0	147	0	0	0	pasta
0	0	1	0	0	0	1	0	93	0	1	rice
4	2	5	4	1	1	3	0	1	281	1	seafood
1	0	5	1	0	0	0	1	0	5	487	soup

Figure 6: Confusion matrix of Food-11 classes using ANN model trained using ResNet-152 features.

ACCEPTED MANUSCRIPT

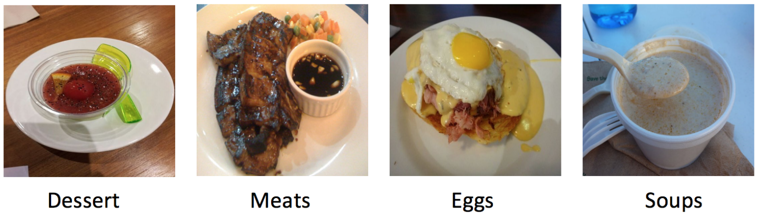


Figure 7: Example of Food-11 classes which are misclassified based on confusion matrix generated from ANN model trained using ResNet-152 features. Images highlight shared characterisitics that could lead to misclassifications.

5.2.2. RawFooT-DB Classification Results

Results listed in Table 13 reveal ResNet-152 features trained with SVM and RBF kernel achieved an accuracy of 99.10% and our ANN also with ResNet-152 99.28% in classifying RawFooT-DB. The results show that deep features efficiently classify isolated texture images across various lighting conditions and further investigation analysing the confusion matrix generated from SVM-RBF model shows that there were a number of classes that experienced misclassifications. For example, several instances were wrongly classified as chickpeas instead of white peas. Investigating the images from both categories, it was clear that there are similarities between shape, colour, and texture as shown in figure 8 and 9. When also investigating the ANN confusion matrix, several white pea instances were also classed as chickpeas and there were also several mango instances classed as apple slice. Figure 9 is an example of image classes that were misclassified using an ANN, chicken breast and milk chocolate. These images showed similar characteristics in colour and texture, similarly hamburger images were classified as salami and further investigation showed very similar texture, colour, and patterns however ResNet-152 features still achieved 0.98 F-measure for hamburgers and 0.99 for salami.

ACCEPTED MANUSCRIPT

Table 13: Classification results using ResNet-152 and GoogLeNet to extract deep features (extracted from RawFoot dataset) with supervised learning classifiers. * denotes highest accuracy achieved.

RawFoot Dataset - Training/Testing Split								
Model	ResNet-152 - pool5				GoogLeNet - cls3_pool			
	Acc (%)	Recall	F1	Kappa	Acc (%)	Recall	F1	Kappa
GNB	82.02	0.82	0.83	0.82	78.42	0.78	0.79	0.78
SVM-RBF	99.10	0.99	0.99	0.99	96.63	0.97	0.97	0.97
SVM-Poly	98.21	0.98	0.98	0.98	96.74	0.97	0.97	0.97
ANN	99.28*	0.99	0.99	0.99	97.04	0.97	0.97	0.97
RF	98.13	0.98	0.98	0.98	94.03	0.94	0.94	0.94

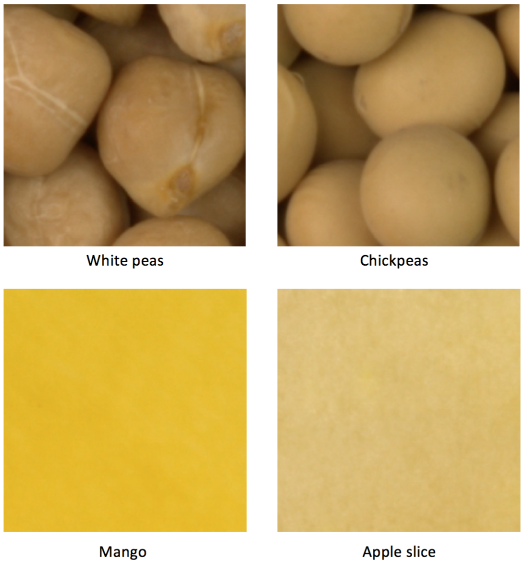
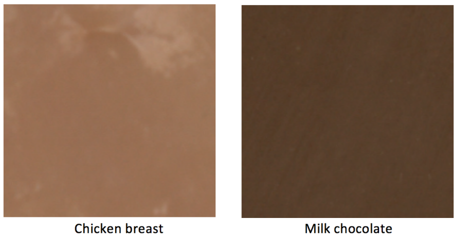


Figure 8: Example of RawFooT-DB classes which are misclassified based on confusion matrix generated from SVM-RBF model trained using ResNet-152 features. Images highlight shared characterisitics that could lead to misclassifications.

.

ACCEPTED MANUSCRIPT


Chicken breast

Milk chocolate

Figure 9: Example of RawFooT-DB classes which are misclassified based on confusion matrix generated from ANN model trained using ResNet-152 features.

For further analysis using RawFooT-DB with ResNet-152 and GoogleNet features, we reordered the food types into 7 groups, vegetables, rice/grains/wheat/seeds, fruits, sweets, breads, meat/fish, and miscellaneous (e.g. coffee, powders, sugar). Figure 10 and 11 show the F-measure of the food texture types rearranged into food groups for ANN and SVM-RBF models. It is clear the from Figure 10 and 11 that there is a decrease in accuracy in ‘meat/ fish’ group. This is evident in Figure 9 as chicken breast can share similar characteristics with other textures such as ’milk chocolate’. Figure 10 and 11 also show decrease in accuracy with chickpeas and white peas due to sharing texture and shape characteristics and this is also evident in Figure 12 using GoogleNet deep features with ANN.

ACCEPTED MANUSCRIPT

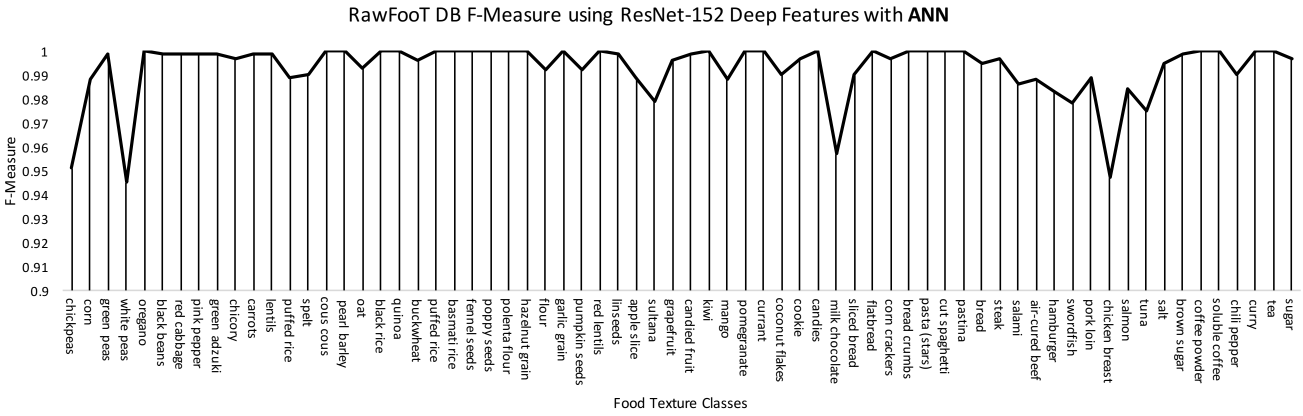


Figure 10: RawFooT-DB F-Measure of reordered classes by major food groups using ResNet-152 features with ANN.

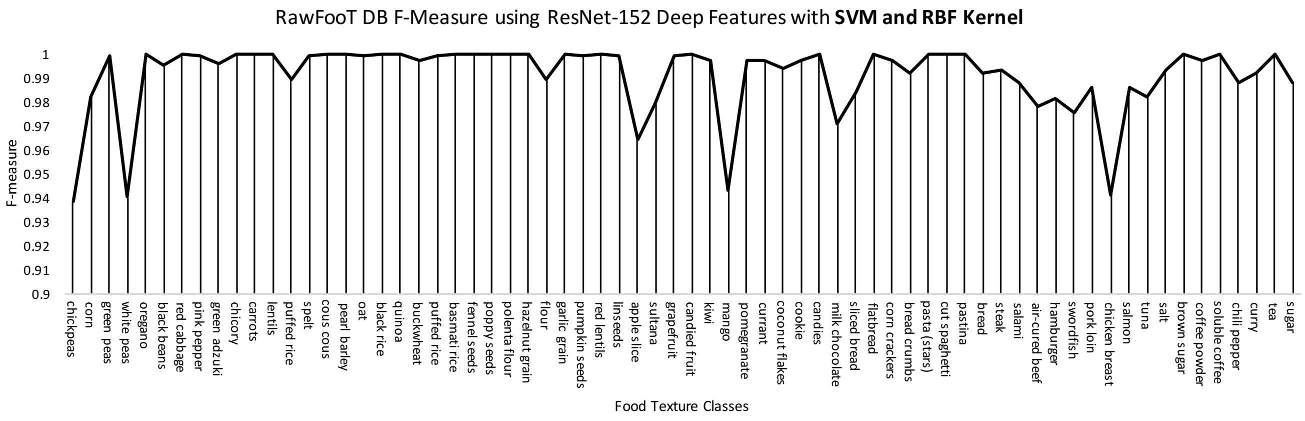


Figure 11: RawFooT-DB F-Measure of reordered classes by major food groups using ResNet-152 features with SVM with RBF kernel.

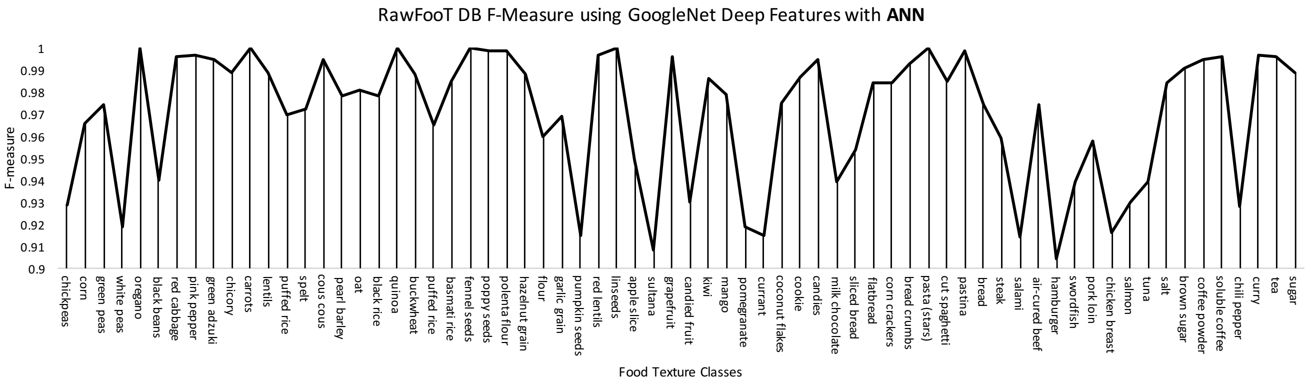


Figure 12: RawFooT-DB F-Measure of reordered classes by major food groups using GoogleNet features with ANN.

5.2.3. Food-101 Classification Results

From previous experiments using Food-5K and Food-11, and RawFooT-DB, ResNet-152 deep features achieved the highest accuracies. We used ResNet-152 features for classifying Food-101, which can be described as fine-grained food image dataset that contains similar food items (i.e. different kind of soups, meats images taken in a free living environment). Results listed in Table 14 show that ANN and SVM-RBF along with ResNet-152 features achieved the highest

ACCEPTED MANUSCRIPT

accuracy across the experiments for Food-101 achieving 64.98%. To train the ANN, Food-101 was partitioned into 75:25, training and testing, with random seed of ’1’ using Weka 3.8.1 (same ANN plug-in used with other experiments for Food-5K, Food-11, and RawFooT-DB). To train the ANN, the learning rate was initially set to 1 with mini-batch gradient descent. For the other classification models we used used Python 2.7.10 with Scikit v0.19. We used Python v2.7.10 and scikit-learn instead of Weka 3.8.1 due to the flexibility of using other libraries and its ease of use when working with larger datasets and also for data analysis. The parameters for the classifiers remained the same as other experiments with Weka as wekaPython contains the same models as scikit-learn. To train the other classifiers using scikit-learn, Food-101 was also split in 75:25 training and testing with a random state parameter of ’1’. Table 14 shows the accuracy, recall, F-Measure, and kappa statistic of using ResNet-152 deep features. The results are much lower than previous experiments with the highest accuracy with 64.18% for ANN and 64.97% for SVM-RBF. The kappa statistic was also generated for ANN and SVM-RBF at 0.64 and 0.65 respectively, which indicates substantial agreement.

Table 14: Classification results using ResNet-152 to extract deep activations (extracted from Food-101 dataset) with supervised learning algorithms. Highest accuracy denoted by *.

Food-101 Dataset - 75:25 training/evaluation				
Model	**ResNet-152 - pool5**			
	Acc	**Recall**	**F1**	**Kappa**
GNB	45.64%	0.46	0.46	0.45
SVM-RBF	64.98%*	0.65	0.65	0.65
SVM-Poly	63.04%	0.63	0.63	0.63
ANN	64.18%	0.64	0.64	0.64
RF	39.33%	0.39	0.38	0.39

ACCEPTED MANUSCRIPT

There were a number of misclassifications that occurred across different classes in Food-101 experiments. Figure 13 and 14 is an example of typical food classes that were misclassified. Misclassifications occured with the steak food class with both the ANN and SVM-RBF. Steak instances were wrongly classified as pork chop, prime rib, and filet mignon using SVM-RBF and ANN, similarly several pork chop instances were classified as steak, prime rib, and foie gras. This may be due to the shared characteristics with shape, texture, and colour. In regards to the desserts, several items were wrongly classified, the panna cotta class was wrongly classified as a cheese cake, and chocolate mousse and the cheese cake class was wrongly classified as a panna cotta, chocolate mousse, chocolate cake, and strawberry shortbread. Further investigation showed that these classes share similar characteristics such as shape and colour which may contribute to them being wrongly classified. Beignets were also wrongly classified as donuts, investigation showed that beignets are very similar to donuts in terms of appearance, texture, colour, and shape, however SVM-RBF trained with ResNet-152 features were still able to achieve an F-measure of 0.77 for beignets.

Figure 15 shows the F-measure for each food class in Food-101 for SVM. For further analysis, we organised the food classes into groups. Images were allocated into groups; (1) breads, pasta, (2) desserts, (3) eggs, (4) fried foods, (5) meats and fish, (6) mixed foods (foods that contained a mixture of foods) and (7) vegetables. Foods were organised into different foods to determine if ResNet-152 features had any inherent advantage for classifying certain food groups. The average F-measure was computed for each group and the vegetable group achieved the highest with an average F-measure of 0.71 using SVM-RBF model, however it should be noted that the vegetable category contained a small number of images in comparison to other groups. In regard to using SVM-RBF model to classify specific food items, the class the achieved the highest F-measure was 'edamame' with 0.98, and further investigation showed that edamame images are very similar as the food item is distinct and there is little variation with

ACCEPTED MANUSCRIPT

the edamame food type and also they are the same shape and colour. The food item that achieved the lowest F-measure was 'steak' with an F-measure of 0.36. Steak food class experienced misclassifications with other food types with other meat classes e.g. pork chop, prime rib, and foie gras due to the similar shape, colour, and texture. In regards to using ANN model, 'edamame' also achieved the highest with 0.97 F-measure and 'steak' was also the lowest with 0.30.

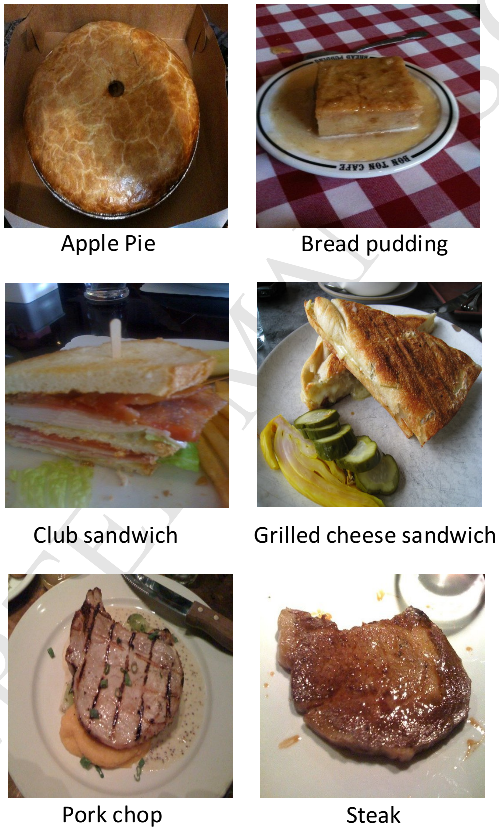


Figure 13: Example of Food-101 classes which were misclassified based on confusion matrix generated from ANN and SVM-RBF models trained using ResNet-152 features. Food classes are on the left experience misclassification with the food classes on the right.

.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

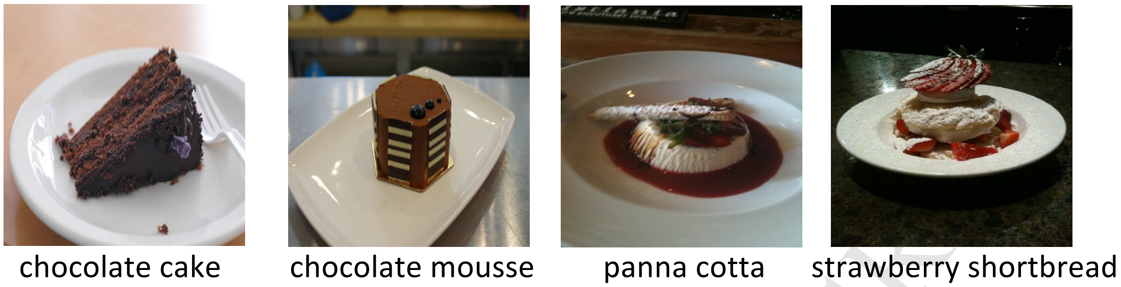


Figure 14: Example of Food-101 dessert classes which were misclassified based on confusion matrix generated using both SVM-RBF and ANN models trained with ResNet-152 features.

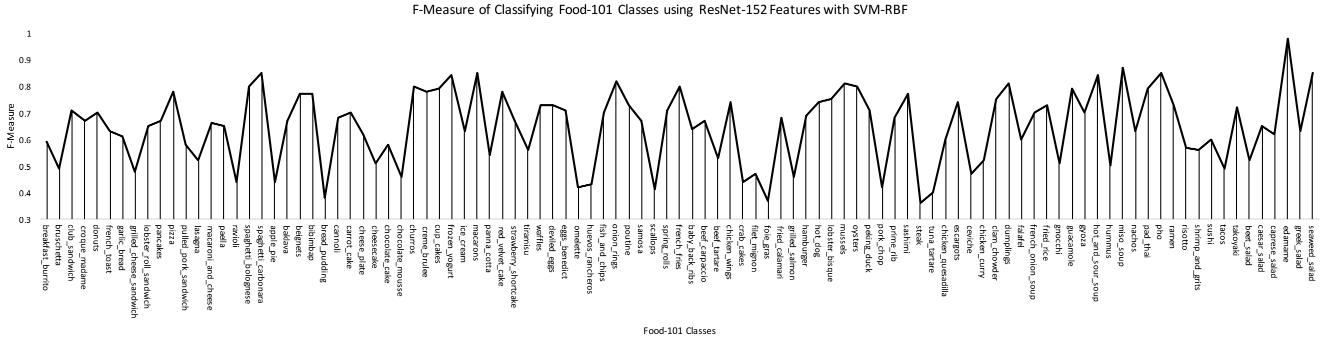


Figure 15: Food-101 F-Measure of reordered classes by major food groups using ResNet-152 features with SVM with RBF kernel.

6. Discussion

In this work we used deep features extracted from pretrained CNNs for food image classification. We compared 2 popular pretrained CNNs, ResNet-152 and GoogLeNet and extracted deep features from layers deep in each CNN architecture to classify Food-5K, Food-11, and RawFooT-DB. For Food-101 we choose to use ResNet-152 deep features as it consistently achieved higher accuracies across other image datasets. We extracted a deep feature vector immediately after the last pooling layer in each architecture for each pretrained CNN for each from various food image datasets. From these experiments, we found that ResNet-152 achieved consistently higher results in Food-5K, Food-11, and RawFoot-DB

ACCEPTED MANUSCRIPT

and because of this ResNet-152 features were used with Food-101. Food-101 is a much more difficult dataset due to the number of classes and variation in images. Many classes contain low in between class variance as many dishes are similar as shown in Figure 13, 14, and 16. From the experiments it was clear that using ResNet-152 is able to achieve high accuracies for Food-5K, Food-11 dataset, RawFoot DB, and moderate accuracy for Food-101.

In regards to Food-5K, the deep features were able to detect food in images with high accuracy across all machine learning classifiers, achieving over 90% accuracy in each experiment. We benchmarked our experiments using the results achieved by the authors of Food-5K and Food-11 datasets who used a fine-tuned GoogleNet [13] and these results in our work suggest that there is potential to achieve high accuracies and performance without the need of fine-tuning pretrained CNNs for certain datasets and problems. Furthermore, due to the nature of Food-5K being a binary decision between food and non-food classes, generic deep features may be sufficient enough to provide adequate generalisation to classify between two classes (i.e. food and non-food).

ANN and SVM-RBF trained with ResNet-152 features achieved the highest accuracies in the majority of Food-5K experiments and the Food-5K ANN and SVM-RBF model was further evaluated by classifying the entire Food-11 dataset for food detection. Results show that our ANN model trained using ResNet-152 features achieved higher food detection accuracy compared to the fine-tuned GoogleNet model in [13] when tested against Food-11 image dataset as stated in Table 15. We also evaluated both our Food/Non-Food SVM-RBF model trained with ResNet-152 and GoogleNet deep features using Food-11 for food detection and results showed that these models achieve marginally higher results compared to other results achieved in also listed in Table 15 [13].

Authors in [13] achieved 83.6% with Food-11 evaluation dataset and in our work ResNet-152 features with ANN achieved 91.34% and 89.99% with SVM-

ACCEPTED MANUSCRIPT

RBF, this is an improvement of 7.74% and 6.39% respectively. For Food-5K, ResNet-152 features achieved 98.8% in classifying Food-5K evaluation dataset and authors in [13] achieved 99.2%. Authors in [13] evaluated their food detection model using all images in Food-11 dataset, we did this also and Table 16 compares our results. ANN and SVM trained with ResNet-152 deep features achieved marginally higher results than achieved in [13] with 97.39% and 97.19% respectively. GoogleNet deep features with ANN also achieved marginally higher results with 97.16% compared to proposed Fine-tuned GoogleNet method in [13].

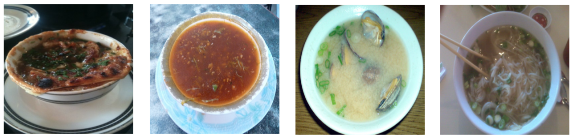


Figure 16: Food image classes from Food-101 that share similar characteristics. Categories from left to right; french onion soup, hot and sour soup, clam chowder, miso soup

Table 15: Method and results comparison using Food-5K and Food-11. * denotes accuracy improvement.

Author	Method	Accuracy	Food Dataset
Singla, et al. [13]	GoogleNet (fine-tuned)	99.2%	Food-5K
Singla, et al. [13]	GoogleNet (fine-tuned)	83.6%	Food-11
This work	ResNet-152 + ANN	98.8%	Food-5K
-	ResNet-152 + ANN	91.34%*	Food-11
-	ResNet-152 + SVM-RBF	89.99%*	Food-11
-	ResNet-152 + SVM-Poly	88.86%*	Food-11

Table 16 also shows GoogleNet features used to detect food images in Food-11. Results show that using GoogleNet features used to train conventional machine learning algorithms is able to achieve higher results than a fine-tuned GoogleNet model in detecting food images in Food-11. These results illustrate

ACCEPTED MANUSCRIPT

Table 16: Results comparison of classifying Food-11 with our Food/Non-Food classification models. * denotes accuracy improvement.

Method	**Num of Food Images Detected**	**Accuracy**
Fine-Tuned GoogleNet [13]	16,127	96.9%
ResNet-152 + ANN	16, 208	97.39%*
ResNet-152 + SVM-RBF	16,176	97.19%*
GoogleNet + ANN	16,171	97.16%*
GoogleNet + SVM-RBF	15,646	94.00%

the convenience of using deep learning with machine learning classifiers through deep feature extraction as the user does not need to use a powerful GPU to quickly train an effective image classification model. Many deep learning packages such as Tensorflow and MatConvNet give users the ability to fine-tune CNNs using CPU, however it has been stated that using a GPU can be around 8 times faster than using a CPU in training a CNN [40].

Food-5K AUC results achieved in this work were close to 1 in validation and evaluation image sets using ANN and RF with both ResNet-152 features and GoogleNet features. However, the validation and evaluation test sets are small in comparison to other popular food image datasets with only 500 in each class for each dataset and therefore more research is needed in classifying a wider range of food images types and image quality. Food-5K training dataset, which was used to train food/non-food models, is also comparatively small with 2500 images in each class and contains limited food image types, therefore further research would need to be completed in training machine learning classifiers with a diverse food image training dataset. Further evaluation was completed using the food/non-food trained models that achieved highest accuracies with Food-5K to classify a new image dataset that combines food images in UNICT-FD889

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

and non-food images Caltech-101, called UNICT-Caltech, which is larger than the validation and evaluation sets provided in Food-5K [52, 53] containing 3583 food images and 9144 non-food images . Results from classifying this dataset are listed in Table 10 and show that with using Food-5K training dataset to train machine learning classifiers is able to achieve a high food accuracy using SVM-RBF achieving 97.50%,.

Further experiments focused on using deep features to classify food texture image items under different illuminations, previous authors of RawFooT DB researched the use of using other popular pretrained CNNs for feature extraction. The experiments presented in this work utilised deep residual network features and GoogLeNet features to classify food images in different lighting settings. Other research that used RawFooT-DB [20] divided the food image classes into illuminant categories. In this work, we evaluated the performance of ResNet-152 features in classifying food texture images across a range of different lighting conditions. Results from using ResNet-152 to train an ANN achieved 99.28% accuracy and and a ROC value of 0.99 and the same features with SVM-RBF achieved 99.10%. More importantly, the use of deep features with supervised machine learning algorithms, from both ResNet-152 and GoogLeNet, are able to generalise between food texture types with great efficiency under different illuminations. Results from RawFooT-DB echos results in early experiments in that ResNet-152 features marginally outperform GoogleNet features even in determining food classes across a number of illuminations. Figure 12 highlights the performance of classifying each texture class in RawFooT-DB using GoogleNet features with ANN, and similar decreases in F-measures are present when compared to ResNet-152 ANN and SVM-RBF in Figure 10 and 11. GoogleNet features also experienced misclassifications with white peas and chick peas, and with several meat textures (salami and hamburger).

Results show that most experiments with RawFooT-DB using both feature types achieved over 90% accuracy (apart from GoogleNet features with Gaussian Naive Bayes, which achieved 78.42%), however ResNet-152 pretrained CNN fea-

ACCEPTED MANUSCRIPT

tures achieves higher accuracy across all machine learning algorithms. This may be due to the increased depth of ResNet-152 CNN in comparison to GoogLeNet CNN and therefore rich detailed features may be extracted from layers deep in ResNet architecture. Pretrained CNN models used in this work were supplied by MatConvNet and experiments in [58] show that ImageNet ILSVRC trained ResNet-152 model outperformed ImageNet ILSVRC trained GoogLeNet Inception model when validating both using ImageNet ILSVRC 2012 validation data using MatConvNet package [58].

There were also several misclassifications between similar food groups with RawFooT-DB. It is worth noting that these food textures that were misclassified are very alike in texture and shape (chickpeas and white peas) and the images used for testing and training are focused on the food texture without the overall food item shape and size as shown in Figure 8 and 9. The use of a texture based classification model trained using deep features may also be very efficient combined with a semi-automation approach to food logging. Future work could enable the user to utilise a polygonal tool to draw around the food item and then a food texture based classifier can you used to predict the food item thus removing much of the complexity and noise of other food and non-food items in the food image. It is clear from the experiments that using pretrained ResNet CNN for deep feature extraction is able to produce feature descriptors that generalise accurately between food texture classes with low in-between variance.

It was revealed that ResNet-152 features continually achieved higher classification accuracy results when compared to GoogleNet therefore ResNet-152 deep features were used to classify Food-101 dataset. The images in Food-101 were not developed in a controlled environment but collated using a social media website (Foodspotting), which were uploaded by users and taken in real world environments (restaurants, at home, cafes, etc.).The images are also taken under illuminations and the dataset contains image quality of the images vary greatly and no bounding box information is provided to help determine where the food

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

items are located in the image. Food-101 contains 101,000 images and 1,000 for each food class, and because of the size of this dataset, we partitioned dataset in training and validation using 75:25 ratio, 75% used for training and 25% used for testing and used a random state of '1' with scikit-learn library. The highest accuracy achieved using ResNet-152 deep features extracted from Food-101 was 64.98% using an SVM with RBF kernel using ResNet-152 features. The full breakdown of results using ResNet-152 to classify Food-101 are located in Table 14. The features extracted from layers deep in CNN architecture provide efficient representations that can be used to classify even the most difficult food image datasets such as Food-101. The quality of food images present in Food-101, in regards to food variation and noise i.e. other non-food items, and unrelated food items, may be a factor in the decrease in accuracy. Comparing the results of Food-101 (101 classes) with RawFooT-DB texture dataset (67 classes) suggest that the class size may not a major determining factor in the decrease in accuracy but the quality of the images used in regards to being truly representative of the class. Results achieved in this work in classifying RawFooT-DB is comparable with results achieved in [20] albeit the authors created small subsets for each lighting condition, while work presented in this paper extracted features from each food class that contains a variety of lighting conditions.

For further comparison, Table 17 lists results achieved in this work with other research that used related deep feature extraction in classifying food image datasets. It is clear from Table 17 and the literature that ResNet-152 deep features echo results achieved with other datasets and other deep feature types [45]. ResNet-152 deep features are able to achieve high classification accuracy in both fine grained datasets such as RawFooT-DB and binary decision datasets e.g. Food/NonFood, however there is a decrease in accuracy when food image datasets with high food variance and noise is present in images as seen in Food-101. A semi-automated approach or segmentation approach could be applied to CNN deep feature classification that allows the user to draw around a food im-

ACCEPTED MANUSCRIPT

age before classification to remove noise, further analysis is needed to evaluate this approach and to measure improvement in accuracy.

ACCEPTED MANUSCRIPT

Table 17: Summary of research using deep feature extraction to classify various food image datasets. **Bold** denotes results achieved in this work. * denotes highest accuracy achieved for Food-5K, Food-11, and RawFooT-DB.

Extraction Model	Accuracy	Food Classes	Dataset
VGG-S [41]	92.47%	2 (Food/NonFood)	RagusaDB
NIN	90.82%	2 (Food/NonFood)	
AlexNet	84.95%	2 (Food/NonFood)	
GoogleNet [42]	94.67%	2 (Food/NonFood)	Based on RagusaDS
	99.01%	2 (Food/NonFood)	FCD
NIN [47]	95.1%	2 (Food/NonFood)	IFD
Singla, et al. [13]	99.2%*	2 (Food/Non-Food)	Food-5K (Evaluation set)
	83.6%	11	Food-11
AlexNet [15]	94.01%	7 (food groups)	PFID
	70.13%	61	PFID
AlexNet [45]	57.87%	100	UEC-FOOD100
AlexNet [45]	70.41%	101	Food-101
AlexNet [45]	78.77%	100	UEC-FOOD100
AlexNet [45]	67.57%	256	UEC-FOOD256
VGG-19 [46]	40.21%	101	UMPC-Food-101
VGG-16 [57]	98.21%	68	RawFooT-DB
VGG-19 [57]	97.69%	68	RawFooT-DB
ResNet-152	**98.8%**	**2 (Food/NonFood)**	**Food-5K (Evaluation set)**
ResNet-152	**99.4%**	**2 (Food/NonFood)**	**Food-5K (Validation set)**
ResNet-152	**91.34%***	**11**	**Food-11 (Evaluation set)**
ResNet-152	**99.28%***	**68**	**RawFooT DB**
ResNet-152	**64.98%**	**101**	**Food-101**

Using CNN deep features to classify food images datasets exceed the per-

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

formance compared to other conventional feature selection methods and has been well documented [45,49,51]. Hand crafted feature selection methods such as SURF, or colour can encounter difficulties when classifying fine-grained classification of food categories as some public food image datasets contain small in-between class differences amongst large number of classes (e.g. Food-101). It has been stated in [51] that deep CNN features should be the first initial method for visual classification tasks due to their high performance in generalising to other datasets as CNNs are trained to be able to learn rich representations from a large number of images. CNNs able to determine complex filters to combine them with other patterns for greater detail. CNNs are able to produce internal image feature representation, which is advantageous when compared to hand crafted feature types such as SIFT, SURF or HOG. In this work, ResNet-152 features are able discriminate effectively between food and non-classes and in classifying high level food groups (Food-11), when compared to other works in [13]. It is clear that using ResNet-152 pretrained model is able to capture relevant image features to enhance the generalisation between fine-grained objects as demonstrated in classifying RawFooT DB in table . ResNet-152 contains 152 layers that combine multiple convolutional and pooling layers to filter important image features and the use of residual connections to train the network produce accurate features which can be highlighted for effective generalisation across other datasets.

It is clear that using CNN features can enhance the accuracy of food image classification when compared to traditional feature extraction methods and this has been observed in other works, for example in [17] SURF and LAB colour features, and Random Forests were used to classify Food-101 dataset and achieved 50.76% accuracy. In [45] an AlexNet model was fine-tuned using food image categories and deep feature extraction was performed after to classify Food-101, and authors achieved 70.41%, which is a significant increase when compared to results achieved in [17]. As well as deep feature extraction, fine-tuning was also used to classify Food-101 and authors in [48] achieved top-1 accuracy of

ACCEPTED MANUSCRIPT

77.4% after 250,000 iterations in training a CNN architecture called 'DeepFood', which is a significant accuracy increase in comparison to [17]. In [49] fine-tuning was also used to classify Food-101 dataset was also used to fine-tune Inception V3 architecture and achieved a top-1 accuracy of 88.28%. Research in [45] also achieved a top-1 accuracy of 65.32% using HOG features, colour values with fisher vectors in classifying UEC-FOOD100, however CNN based features extracted from a modified AlexNet model with a linear SVM achieved an increased accuracy of 78.77%. For UEC-FOOD256 dataset, work presented in [50] achieved a top 1 accuracy of 50.1% using HOG features and colour features with Fisher Vector representations and the same authors in later research [45] utilise deep CNN features extracted from a modified AlexNet and achieved a top 1 accuracy of 67.57% in also classifying UEC-FOOD256 dataset. For Raw-FooT DB food texture dataset experiments were completed in classifying food textures under various lighting conditions, authors compared traditional feature extraction techniques with CNN based features, and results show that OCLBP and Gabor features achieved 95.9% and 96.2% accuracy respectively with deep CNN features achieving 98.2% accuracy [20]. From the literature it is clear that using CNN deep feature extraction and fine-tuning can achieve superior results in regards to food image classification.

7. Limitations & Future Work

There are a number of limitations associated with this study which could be addressed in future works, for example, an expansive dataset could be developed under a controlled environment that is representative of a broad range of food items. This dataset could be used with the methods outlined in this work and compared with similar works. This would give a clear indication of the true performance of using deep feature extraction with machine learning algorithms. Also, a comprehensive study could be completed by fine-tuning a range of CNNs on food datasets and comparing performance using the same pre-trained CNN models for deep feature extraction. Further experiments can also

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

be completed by comparing deep features extracted from different layers within a CNN architecture to find what layer is more suitable for generalising between different food classes. In regards to overfitting, particularly for Food-101, future works could include using 10-fold class validation instead of using a 75:25 train/testing split. This would give a clearer indication of the performance of using deep features from ResNet-152 and GoogLeNet. Some of the experiments in this work achieved high accuracies, especially for Food/Non-Food classification experiments, however it is important to note that the amount of images contained in Food-5K are relatively small in comparison to other datasets e.g. Food-11 or Food-101. Further experiments need to be completed in detecting food/non-food in larger food image datasets in using off the shelf deep features.

For RawFoot-DB we used the training and test split provided by authors in [20, 42], however the authors of RawFooT DB in [20] created subsets of each category, which were based on lighting condition type. In this work, our aim was to classify food textures across different lighting conditions, however in future work we would follow the same procedures described in [20] and use ResNet-152 features for further comparison. Also authors of [17] allocated a testing split that contained images that contained little noise and representative of each class, however in our work Food-101 extracted features were shuffled using random seed '1' and random state '1' to determine the classification performance of ResNet-152 features when used with images with high level of noise. In future works, we will further evaluate ResNet-152 features following the partition procedure described in [17].

Future work could incorporate hierarchical classification using pretrained CNN features in which a classifier will be used to determine food and non-food images, another classifier will be appended that determines major food groups, and finally a further classifier will used after to determine low level food item. Further experiments with the parameters of machine learning models could also be changed in order to determine the optimal parameter settings to achieve a

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

high classification accuracy. The presence of noise in the food image datasets may also affect the accuracy, in order to mitigate these issues, a semi-automated approach could be adopted by using a polygonal tool to draw around the food portion and to ultimately segment the food item. Classification models could then classify the segmented food portion in order to promote accuracy. Other computer vision segmentation approaches could be researched and combined with methods described in this work. For future evaluation, we would also input random noise as feature vectors for trained classifiers to determine food classes and analyse the output and performance. The use of machine learning models using pretrained CNN deep features also have the potential of being using in mobile health solutions. Much research has been dedicated to understanding a person's diet by determining what major food groups they consume daily [2,5]. This research has showed that this process can be automated using deep features extracted from residual CNNs for high food classification accuracy. From this research, it is clear that ResNet-152 deep features is able to distinguish between high-level food categories such as Food/Non-food and echoes other related research in this area. In comparison with other works, ResNet-152 deep features outperforms other CNN deep features such as GoogleNet in distinguishing between fine-grained food texture classes in RawFooT DB and is comparable with other related works [20]. ResNet-152 features encountered some difficulty in classifying Food-101 classes, however this may be due to the images containing noise in the form of high colour intensities and multiple foods in the same image, however a reasonable accuracy of 64.98% was achieved. In Food-11 food group classification, deep GoogleNet features were able to achieve high accuracy result when compared to research presented in [13] which used a fine-tuned GoogleNet, which shows that a combination of conventional machine learning classifiers combined with CNN deep features have the ability to outperform fine-tuned models.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

8. Acknowledgements

Thank you to Department of Education and Learning Northern Ireland (DELNI) for funding this research.

9. References

1. M. Di Cesare et al., “Trends in adult body-mass index in 200 countries from 1975 to 2014: A pooled analysis of 1698 population-based measurement studies with 19.2 million participants,” The Lancet, vol. 387, no. 10026. pp. 13771396, 2016.

2. Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, “Food image analysis: Segmentation, identification and weight estimation,” Proc. - IEEE Int. Conf. Multimed. Expo, 2013.

3. T. Lehnert, D. Sonntag, A. Konnopka, S. Riedel-Heller, and H.-H. Knig, “Economic costs of overweight and obesity,” Best Pract. Res. Clin. Endocrinol. Metab., vol. 27, no. 2, pp. 105115, 2013.

4. C. M. Wharton, C. S. Johnston, B. K. Cunningham, and D. Sterner, “Dietary Self-Monitoring, But Not Dietary Quality, Improves With Use of Smartphone App Technology in an 8-Week Weight Loss Trial,” J. Nutr. Educ. Behav., vol. 46, no. 5, pp. 440444, 2014.

5. M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, “A food recognition system for diabetic patients based on an optimized bag-of-features model,” IEEE J. Biomed. Heal. Informatics, vol. 18, no. 4, pp. 12611271, 2014.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

6. G. M. Farinella, M. Moltisanti, and S. Battiato, “Food recognition using consensus vocabularies,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9281, pp. 384392.

7. H. He, F. Kong, and J. Tan, “DietCam: Multiview food recognition using a multikernel SVM,” IEEE J. Biomed. Heal. Informatics, vol. 20, no. 3, pp. 848855, 2016.

8. N. Martinel, C. Piciarelli, C. Micheloni, and G. L. Foresti, “A Structured Committee for Food Recognition,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, vol. 2015February, pp. 484492.

9. ImageNet Large Scale Visual Recognition Competition (ILSVRC)”, Image-net.org, 2017. [Online].
Available: http://www.image-net.org/challenges/LSVRC/. [Accessed: 16-Sep- 2017].

10. A. Krizhevsky, I. Sutskever, and G. E. Hinton,“ImageNet Classification with Deep Convolutional Neural Networks,” Adv. Neural Inf. Process. Syst., pp. 19, 2012.

11. N. Tajbakhsh et al., “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?,” IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 12991312, 2016.

12. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 512519.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

13. A. Singla, L. Yuan, and T. Ebrahimi, "Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model," in Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa 16, 2016, pp. 311.

14. H. , K. Aizawa, and M. Ogawa, "Food Detection and Recognition Using Convolutional Neural Network," ACM Multimed., no. 2, pp. 10851088, 2014.

15. M. Farooq and E. Sazonov, "Feature Extraction Using Deep Learning for Food Type Recognition" in Bioinformatics and Biomedical Engineering: 5th International Work-Conference, IWBBIO 2017, Granada, Spain, April 26–28, 2017, Proceedings, Part I, I. Rojas and F. Ortuo, Eds. Cham: Springer International Publishing, 2017, pp. 464472.

16. Y. Kawano and K. Yanai,"Food Image Recognition with Deep Convolutional Features" ACM Int. Jt. Conf. Pervasive Ubiquitous Comput., pp. 589593, 2014.

17. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 - Mining discriminative components with random forests," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8694 LNCS, no. PART 6, pp. 446461.

18. Y. Matsuda, H. Hoashi, and K. Yanai,"Recognition of multiple-food images by detecting candidate regions" in Proceedings - IEEE International Conference on Multimedia and Expo, 2012, pp. 2530.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

19. Y. Kawano and K. Yanai,“Automatic expansion of a food image dataset leveraging existing categories with domain adaptation,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 8927, pp. 317.

20. C. Cusano, P. Napoletano, and R. Schettini, “Local angular patterns for color texture classification,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9281, pp. 111118.

21. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, “Large-scale video classification with convolutional neural networks,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 17251732.

22. C. Szegedy et al., “Going deeper with convolutions,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 7-12-2015, pp. 19.

23. A. Vedaldi and K. Lenc, MatConvNet, in Proceedings of the 23rd ACM international conference on Multimedia - MM 15, 2015, pp. 689692.

24. ”Image Category Classification Using Deep Learning”, Mathworks.com, 2017. [Online]. Available: https://www.mathworks.com/examples/matlab-computer-vision/mw/vision_product-DeepLearningImageClassificationExample-image-category-classification-using-deep-learning. [Accessed: 18- Sep- 2017].

25. K. He, X. Zhang, S. Ren, and J. Sun,“Deep Residual Learning for Image Recognition” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770778.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

26. K. Simonyan and A. Zisserman,“Very Deep Convolutional Networks for Large-Scale Image Recognition,” Int. Conf. Learn. Represent., pp. 114, 2015.

27. 28. R. G. Pontius and M. Millones, “Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment,” Int. J. Remote Sens., vol. 32, no. 15, pp. 44074429, 2011.

28. “Weka 3 - Data Mining with Open Source Machine Learning Software in Java,” Cs.waikato.ac.nz, 2017. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/index.html. [Accessed: 18- Sep- 2017].

29. “Waikato Environment for Knowledge Analysis (WEKA),” Weka.sourceforge.net, 2017. [Online]. Available: http://weka.sourceforge.net/packageMetaData/wekaPython/Latest.html. [Accessed: 18- Sep- 2017].

30. “Java (convolutional or fully-connected) neural network implementation,” GitHub, 2017. [Online]. Available: https://github.com/amten/NeuralNetwork/releases/tag/v1.1. [Accessed: 18- Sep- 2017].

31. H. Zhang,“The Optimality of Naive Bayes,” Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004, vol. 1, no. 2, pp. 16, 2004.

32. I. H. Witten, E. Frank, and M. a Hall, Data Mining: Practical Machine Learning Tools and Techniques. 2011.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

33. T. Malisiewicz, A. Gupta, and A. A. Efros,“Ensemble of exemplar-SVMs for object detection and beyond,” in Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 8996.

34. C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 415-425, 2002.

35. G. James, D. Witten, T. Hastie, and R. Tibishirani, An Introduction to Statistical Learning. 2013.

36. J. P. Mueller, et al,“Hitting Complexity with Neural Networks,” Machine Learning for Dummies, Hoboken, New Jersey, Wiley, 2016, ch. 16, pp.279-290.

37. L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 532, 2001

38. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346359, 2008.

39. T. Ojala, M. Pietikinen, and T. Menp, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971987, 2002.

40. “GPU vs CPU in Convolutional Neural Networks using TensorFlow — Relink”, Relink, 2017. [Online]. Available: https://relinklabs.com/gpu-vs-cpu-in-convolutional-neural-networks-using-tensorflow. [Accessed: 19-Sep- 2017].

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

41. H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, Automatic diet monitoring: a review of computer vision and wearable sensor-based methods., Int. J. Food Sci. Nutr., pp. 115, 2017.

42. “RawFooT DB”, Projects.ivl.disco.unimib.it, 2017. [Online]. Available: http://projects.ivl.disco.unimib.it/minisites/rawfoot//. [Accessed: 19- Sep-2017].

43. F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella, Food vs Non-Food Classification, in Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa 16, 2016, pp. 7781.

44. E. Aguilar, et al. ”Exploring Food Detection using CNNs,”arXiv:1709.04800v1 [cs], Sept 2017.

45. K. Yanai and Y. Kawano, Food image recognition using deep convolutional network with pre-training and fine-tuning, in 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2015, pp. 1 - 6

46. X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, Recipe recognition with large multimodal food dataset, in 2015 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2015, 2015.

47. Kagaya and K. Aizawa, Highly Accurate Food/Non-Food Image Classification Based on a Deep Convolutional Neural Network BT - New Trends in Image Analysis and Processing – ICIAP 2015 Workshops: ICIAP 2015 International Workshops, BioFor, CTMR, RHEUMA, ISCA, MADiMa, SBMI, and QoEM, Genoa, Italy, September 7-8, 2015, Proceedings, V.

Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Cham: Springer International Publishing, 2015, pp. 350357.

48. 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 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, vol. 9677, pp. 3748.

49. H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, Food Image Recognition Using Very Deep Convolutional Networks, in Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa 16, 2016, pp. 4149.

50. Kawano Y, Yanai K ,“FoodCam-256: A large-scale real-time mobile food recognition system employing high-dimensional features and compression of classifier weights”., MM 14, 2014, 761-762.

51. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 512519.

52. L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst., vol. 106, no. 1, pp. 5970, 2007.

53. G. M. Farinella, D. Allegra, and F. Stanco, A benchmark dataset to study the representation of food images, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture

ACCEPTED MANUSCRIPT

Notes in Bioinformatics), 2015, vol. 8927, pp. 584599.

54. “Welcome to Python.org”, Python.org, 2017. [Online]. Available: https://www.python.org/. [Accessed: 24- Nov- 2017].

55. “scikit-learn Machine Learning in Python”, Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/. [Accessed: 24- Nov- 2017].

56. S. Boseley, ”Global cost of obesity-related illness to hit $1.2tn a year from 2025”, the Guardian, 2017. [Online]. Available: https://www.theguardian.com/society/2017/oct/10/trea obesity-related-illness-will-cost-12tn-a-year-from-2025-experts-warn. [Accessed: 24- Nov- 2017].

57. C. Cusano, P. Napoletano and R. Schettini, ”Evaluating color texture descriptors under large variations of controlled lighting conditions”, Journal of the Optical Society of America A, vol. 33, no. 1, p. 17, 2015.

58. ”Pretrained CNNs - MatConvNet”, Vlfeat.org, 2018. [Online]. Available: http://www.vlfeat.org/matconvnet/pretrained/. [Accessed: 11- Feb- 2018].

59. Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248 - 255.

ACCEPTED MANUSCRIPT