AMERICAN UNIVERSITY OF BEIRUT

Studying Children Food Exposure and Food Consumption using Deep Learning

by
Zoulfikar Ali Nasser Shmayssani

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
December 2021

AMERICAN UNIVERSITY OF BEIRUT

Studying Children Food Exposure and Food Consumption using Deep Learning

by

Zoulfikar Ali Nasser Shmayssani

Approved by:

Dr. Shady Elbassuoni, Associate Professor Computer Science	Advisor
Dr. Hala Ghattas, Associate Research Professor Health Sciences	Member of Committee
Dr. Hazem Hajj, Associate Professor Electrical and Computer Engineering	Member of Committee
Dr. Fatima Abu Salem, Associate Professor Computer Science	Member of Committee

Date of thesis defense: December 10, 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Shmayssani Zoulfikar Ali Nasser
Last First Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

1/5/2022

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I am deeply grateful to my thesis advisor Prof. Shady Elbassuoni for his excellent guidance, patience, and persistent help. Without him, this thesis would not have been possible.

My sincere thanks must also go to the members of my thesis committee: Prof. Hazem Hajj, Prof. Fatima Abu Salem, and Prof. Hala Ghattas who accepted to serve on my committee. They provided helpful suggestions to improve my work.

Finally, I would like deeply to thank my parents for supporting me throughout all my studies. They were always encouraging me with their best wishes.

An Abstract of the Thesis of

Zoulfikar Ali Nasser Shmayssani for Master of Computer Science
Major: Computer Science

Title: Studying Children Food Exposure and Food Consumption using Deep Learning

Children's eating behaviours is one of the main pillars of a healthy life. Recent studies show that eating unhealthy food is highly associated with many chronic diseases including diabetes, obesity, and cancer. Such dietary habits are often shaped by complex factors influenced by the children's home, school, and neighborhood environments. However, studying the eating behaviours of children and analyzing the factors affecting them is currently done using traditional questionnaire-based methods, which often suffers from recall and bias issues. In this thesis, we developed a comprehensive approach to study children food exposure and food consumption using deep learning. Our approach takes as input a set of images captured automatically using wearable cameras and that contain any exposure to food, including actual food items, food outlets, and food advertisements. Our approach then relies on a series of deep learning models to 1) classify food exposure images into one or more of the above-mentioned classes, and 2) to predict the healthiness of any food items consumed in all the images, using the NOVA classification system as a measure of healthiness. To be able train all of these models, we relied on crowdsourcing to generate the training data. First, we built the food exposure dataset that contains 3,560 images that belong to the different food exposure classes. Then, the NOVA dataset that was labeled by Tunisian expert dietitians contains 3,728 food items labeled by bounding boxes that belong to the different NOVA groups. After training our models, we evaluated them on the testing datasets. The food exposure models achieved an average f1-score of 0.96. The food item detection model achieved a mAP@0.5 of 0.90. Finally, the average f1-score of the NOVA classification model was 0.86. After validating our models, we deployed them in a real world case study in Greater Tunisia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Estimating the healthiness of food items consumed by children has become an important goal in the last few years because the eating behavior of a child affects his/her health. Eating unhealthy food may result in many health problems such as diabetes, lack of energy and even cancer [2]. This eating behavior is usually driven by many factors that are related to the community and the neighborhood environments of the children [3]. Children are exposed in their daily lives when going to and from their schools to different types of food exposure that have an influence on the children's food preferences and eating behaviors. This food exposure includes food outlets, food advertisements, and food consumption. Therefore, it is important to classify the food exposure scenes into a hierarchy of classes which help health researchers to analyze the relations between the detected food exposures and the dietary and eating behaviours of children. However, this classification and analysis is typically being done using traditional data

collection tools and recall-based interviews with professional dietitians which has many limitations including inaccuracies and poor recall [4, 5].

In this thesis, we propose an approach that is based on a series of deep learning models that aim to automatically classify images captured through wearable cameras into different food exposure categories, detect food items in those images, and asses their healthiness. First, we build a food exposure classification model that aims to classify the images into one or more of the following food exposure categories: Food Consumption, Food Outlet, Food Advertisement. The Food Consumption classification model comes after, and it aims to classify the images into one or more of the following categories: Personal Food Consumption and Others Food Consumption. After building the food exposure and consumption typology models, we develop a food healthiness prediction approach that is based on an accurate food item detection model followed by a food healthiness model that classifies the detected food items into different groups based on their processing level. After building and validating these models, we deployed them in a real-word case study in Tunisia that aims to study the eating behaviors of children, and explore the factors that affects their food choices.

1.1 Challenges

Building a supervised computer vision deep learning model requires a big number of good quality labeled images. The first challenge that we faced is the lack

of labeled datasets that meet our requirements. There are no available public datasets that contain labeled images with a food exposure typology (Food outlets, food advertisements, and food consumption). Also, there are no datasets that contain labeled images with the categories of personal food consumption and others food consumption. To solve this issue, we used crowdsourcing to build and label different datasets for training our models. In addition to that, we used transfer learning which is defined by Ranaweera et al. [6] as a technique that uses a model that is trained on a certain task to do a different task by transferring the learnt knowledge. Some of the advantages of using transfer learning that are discussed in [6] are: training a model on a new task using a small amount of labeled data and providing an optimized starting point, a faster training, a higher accuracy, and a generalization for unseen classes.

The second challenge was building multi-label classification models that aim to classify each image into one or more classes at the same time. In [7], researchers discussed some of the main challenges that face this type of models, including unbalanced data and the high dimensionality of the label space. These problems result in a low precision and recall, especially when there are no enough training instances that belong to a combination of classes [7]. We tackled this challenge by oversampling the images of the underrepresented classes using data augmentation techniques such as vertical and horizontal flips, rotations, brightness variations, etc.

Building a generic and accurate food item detection model is a challenging

problem in computer vision. Many researchers have attempted to do this, however, they were using datasets that contain homogeneous and pre-processed food-related images. In this thesis, our dataset contains food consumption images that are taken from wearable cameras, which makes the task of detecting food items more challenging. The captured images are not homogeneous and they contain food items with various types, sizes, shapes, and distances from the camera. In addition to that, these images contain various non-food related objects since the images were captured in different places such as schools and houses. We overcame those challenges by building an optimized YOLOv3 model [8] that applies the detection on three different image scales. In addition, our model is based on a backbone that was trained on a large food-related dataset, which helps in detecting only the food related objects in an image.

Finally, food-item healthiness estimation is a challenging task that was addressed using multiple approaches. Most of these approaches however use calories and volume estimation techniques, which require images that are captured by cameras from predefined distances and angles. Also, the images should contain reference objects. This approach does not work on our data because the images were captured using wearable cameras. This resulted in images with various angles and distances to the food items. In this thesis, we propose to build a multi-label classification model that predicts the healthiness of food items using the NOVA classification system [9].

1.2 Objectives and Contributions

In this thesis, we aim to build a novel approach that is based on a hierarchy of deep learning models that are able to classify food exposure images in the wild into different food exposure categories, as well as detect the food items and asses their healthiness qualitatively. To achieve our aim, we first built a multi-label food exposure model that classifies each image into one or more of the following categories: Food Consumption, Food Outlet, and Food Advertisement. The Food consumption multi-label model comes after, and it classifies the Food Consumption images into one ore more of the following categories: Personal Food Consumption and Others Food Consumption. After building the first two classification models, we developed a generic food item detection model that is based on an optimized version of YOLOv3 model [8] and we showed that it outperforms the standard version of the model on our task. Finally, and on the contrary to most of the approaches that were proposed previously for food healthiness prediction and that use quantitative techniques that are based on volume and calories estimation, we proposed an approach to qualitatively asses the healthiness of of food items based on the NOVA classification system [9], and which classifies food items into four groups based on their processing levels. For each of the four models, we also built training data that meets the model's requirements. Since the datasets that we labeled are not very large, we used transfer learning to train our models which led to a better performance and

accuracy. These models were deployed in a case study in Tunisia involving school children, and our models have shown to be very beneficial in classifying food exposure images captured through wearable cameras into their respective classes, as well as in estimating the healthiness of food items consumed in this images.

Our contributions in this thesis can thus be summarized as follows:

1. We built a food exposure multi-label classification model and a food consumption multi-label classification model.

2. We developed an accurate and generic food item detection model that is based on an optimized version of YOLOv3 which outperformed the standard version of the model.

3. We built a qualitative healthiness prediction multi-label model that classifies food items into the different NOVA groups.

4. We created the needed datasets for each model using crowdsourcing: Food Exposure, Food Consumption, and NOVA datasets.

1.3 Thesis Outline

This thesis is organized as follows: Chapter 2 surveys literature about food items detection, recognition, and healthiness prediction using deep learning approaches. Chapter 3 describes the Food Exposure and Consumption Typology models that aim to classify the food related images into a hierarchy of classes. In Chapter 4, we

explain the approach that we used for food item healthiness prediction, which is based on an optimized YOLOv3 object detection model and a NOVA healthiness prediction model. For each of the models, we present the datasets that we built and used in training and testing the models.In chapter 5, we describe the case study that employed our developed models to study food exposure among school children in Greater Tunisia, and predict the healthiness of food items consumed by those children. Finally, Chapter 6 concludes the thesis and suggests future work.

Chapter 2

Literature Review

There is a wealth of work on the topic of dietary and food analysis using machine learning. These related works use different approaches for food items detection, recognition, and healthiness prediction.

2.1 Food Items Detection and Recognition

Most of the machine learning work related to food and dietary analysis require algorithms and models for food items detection, recognition, and segmentation. In [10], the authors propose a CNN architecture that is based on the ResNet-5 pre-trained model that is used to extract the features from fast food images. The extracted features are then used to train a multiclass Support Vector Machine (SVM) classifier that classifies fast food images into 10 classes. This model achieved 94% accuracy on the PFID dataset [11]. Liu et al. [12] proposed a

deep learning approach based on CNNs that accurately classifies food images that are captured in the real world. Others modified and optimised the Inception model architecture [13] by adding convolutional layers which increased the depth of the Inception model and decreased its the dimension at the same time. Their model outperformed all of the previous experiments that were done on UEC-256 dataset [14] with 94.6% top-5 accuracy and Food-101 dataset [15] with 93.7% top-5 accuracy.

Aguilar et al. [16] developed a framework that targets the problem of automatic food tray analysis in restaurants. Their system is based on CNNs, and it consists of food localization, recognition, and segmentation. The first part of the framework is a food segmentation model that is based on Fully convolutional networks (FCNs), and it aims to separate food-related items from the background image (the tray). After that, Moore-Neighbor tracing algorithm is applied to the binary image, which is predicted by the FCN model, to detect the exterior boundaries of the food items and then generate the corresponding bounding boxes. The second part does food items object detection using YOLOv2 model. The results of the two parts of the framework are combined to decrease false positives detection. They have achieved a 0.911% F2 score. Bolanos et al. [17] proposed another approach for a generic simultaneous food Localization and recognition. First, they trained the CNN GoogleNet architecture [13] to distinguish between food and non-food images and they reached a 95.64% testing accuracy. Second, they modified the previous architecture by adding Global Average Pooling

(GAP) layer that aims to generate heat maps of foodness probabilities. Finally, bounding boxes are generated for the regions with a probability above a certain threshold. After detecting the food items, they fined tuned GoogleNet architecture to classify them into their types. This approach was trained and tested on UECFood256[14] dataset and resulted in a precision of 54.33% , a recall of 50.86% , and an accuracy of 36.84%. In addition to that, they built the EgocentricFood dataset, which contains food consumption images that are taken from wearable cameras. Their approach was also tested on this dataset, however they achieved a precision of 17.38% , a recall of 8.72% , and an accuracy of 6.41%. The authors explained that their approach faced problems on the EgocentricFood dataset because the images in this dataset were taken from a lateral point of view where the quality of the images is lower and the food items are far away from the camera wearer. Also, there are some food items that are difficult to distinguish from the non-food objects especially when big parts of the food items are occluded.

2.2 Food Items Healthiness Prediction

Predicting the healthiness of food using the corresponding food image is a challenging problem in computer vision. Current work mainly uses food calories estimation for assessing the healthiness of the food. Liang et al. [18] proposed a calorie estimation system that takes two images as an input: a top and a side view of the food dish that include on its side a coin, which is used as a calibration

object. They used Faster R-CNN for detecting food items using bounding boxes. After that, they applied image segmentation on the detected food items for background removal using GrabCut algorithm. The segmented images are then used to estimate their volume and mass, which are used to estimate the number of Kcal per food item. Their system achieved a 93.0% mean Average Precision(mAP), and the volume estimation error did not exceed ±20%. The authors in [19] developed the Im2Calories system that estimates the number of calories per food dish. They started by training a GoogLeNet model on Food101 multi-labeled dataset and they achieved an overall of 0.5 mAP. Then, they used DeepLab system for semantic image segmentation which allows them to localize food items and segment them. Using the voxel representation and the segmentation mask of food items, they estimated the volume of each food item, and consequently predict the number of calories using the calorific density of each kind of food. The authors, however, faced the problem of insufficient calorie-annotated dataset and they could not do sufficient evaluations because the texture properties and the color of the images in their dataset are different from the ones of real food images. Similarly, in [20], the authors build an AI system that is able to estimate the nutrients intake in hospitals. They built a dataset of 660 images by setting up a table that contains a camera on the top with a specific distance from the food items. In addition to that, they created a database that contains the recipes and the nutrients information of the consumed meals. They used a Multi-Task Fully Convolutional Network model for image segmentation that aims to estimate the

food items volumes which helped in estimating the nutrient intakes based on the created database. Their system has a 15% estimation error.

Gao et al. [21] introduced MUSEFood, which is a food volume estimation system that is different from all of the previous volume estimation methods. Their proposed system does not require any training using food images with their corresponding volume information, and in addition eliminates the need to place a reference object of a known size when capturing the images. Instead, they used microphones and speakers to calculate the vertical distance from the camera to the food item, which helps in estimating the actual volume of the food and thus estimating the number of calories. Also, they used FCN for food images segmentation and they got mIoU of 0.92. Their experiments show that MUSEFood system outperforms all of the state-of-the-art methods with an error of +2.7% for food in plates, and -0.27% for food on bowls [21]. This error however increases for food items with irregular shapes.

Overall, using volume and calories estimation approaches for assessing the healthiness of food items has many limitations including 1) the fact that the pictures of the food items should be captured from specific angles, 2) the need for reference objects, which are used in volume estimation, should be placed in all of the images, 3) training these models typically require a large number of annotated images for each type of food items, and 4) there should be a specific predefined database that contains the nutrients information about the food items that exist in the images dataset.

Sudo et al. [22] proposed a different healthiness prediction approach that is based on a feature extraction deep learning model that is followed by ranking algorithm. First, they built a dataset of 850 food meals images that are taken from a top-view. These images were ranked by registered dietitians based on the healthiness of the whole meal from best to worst. Second, they built a feature extraction model that uses a CNN followed by a pyramid scene parsing network (PSPNET) [23], which outputs pixels-based feature maps. The extracted features are used as an input to the ranking algorithm that uses another CNN. The authors found the correlation coefficient between the dietitians judgement and the ground truth rank that is based on the nutritional measurements of the meals to be 0.73, which is low. Their system achieved an accuracy of 83.6% for ranking pairs of test images. Therefore, this approach did not result in a high enough accuracy because assessing the healthiness of food meals by ranking them from best to worst without a specific criteria is not highly correlated with the ground truth healthiness of the food items.

In this thesis we built a healthiness prediction system that is composed of two models: Food Item Detection model, and the NOVA classification model. Our approach is different from the previous methodologies discussed above, where we propose to build a food item detection model that generically detects the food items of different shapes, sizes, and types. Our model is based on an optimized version of YOLOv3 that is able to detect food items on three different image scales with a high accuracy. In addition, most of the previous approaches used

a quantitative systems that measure the healthiness of food items based on the number of calories. Instead of that, we used the NOVA classification system that aims to qualitatively asses the healthiness of food items by classifying them into four groups according to the level of processing they have undergone. This approach overcomes the limitations of volume and calories estimation methods such as the need to label large number images for each type of food items using masks and polygons. Also, the model can be applied on images that are taken from wearable cameras without any predefined angles.

Chapter 3

Food Exposure and Consumption Typology Models

The aim of the Food Exposure and Consumption Typology models is to classify food-related images captured by wearable cameras into a hierarchy of food exposure and food consumption classes. Our proposed approach is based on two models as shown in Figure 3.1. The first model is the multi-label Food Exposure model that classifies each image into one or more of the following classes: Food Outlet, Food Consumption, and Food Advertisement. The Food Consumption category consists of images that contain food items that are being consumed or about to be consumed. The Food Outlet Category includes images that contain a food outlet such as a supermarket, a shop, a restaurant, a kiosk, a cafe, etc. Finally, the Food Advertisement category includes any ads that are related to food such as billboards, storefront ads, etc.

The second model is the Food Consumption model that aims to further classify the Food Consumption images into Personal Food Consumption, Others Food Consumption, or both of the categories. An image belongs to Self Food Consumption if it contains food items that the person wearing the wearable camera is obviously consuming or is about to consume. An image belongs to Others Food Consumption if it contains other people consuming food or about to consume food. An image belongs to both of categories if the person wearing the wearable camera is eating food with other people.

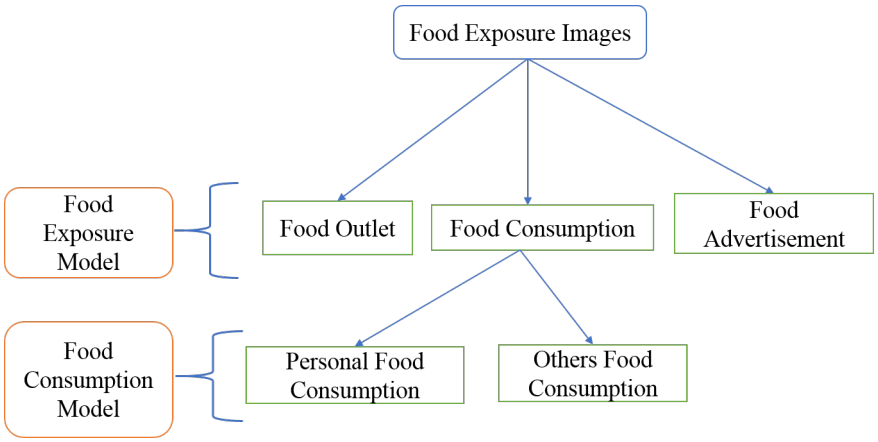

Figure 3.1: Food Exposure Models Typology

3.1 Datasets

3.1.1 Children Trajectory Dataset

The dataset that we used in this thesis was generated by Yorgo et. al [24] in Greater Tunis. The dataset was collected from wearable cameras of 265 children

from 29 schools. After prepossessing the images, a machine learning model was used to filter the Food Exposure related images only such as images containing food outlets, food advertisements, and food consumption. Using the filtered food exposure dataset, which contains 30,170 images, we sampled 3,560 unique and representative images that belong to the different classes we have. We used Labelbox platform [25] for crowdsourcing and labelling the sampled images to build our Children Trajectory dataset. First, we trained a team of five annotators, who work at Labelbox as full time labelers, on the labeling task and we provided them with the needed instructions. The task was that for each image, they should select one or more of the following categories: Personal Food Consumption, Others Food Consumption, Food Outlet, and Food Advertismnet. We ensured the quality of the annotations by reviewing the labeled images through a voting system, where the incorrect labels were given down-votes and then they were corrected by the labelers. Moreover, there was a direct communication with the labelers through a shared document where they can ask about the ambiguous images. Table 3.1 shows the distribution of the labeled images over the classes.

Category	**Count**
Personal Food Consumption	1600
Others Food Consumption	340
Personal and Others Food Consumption	940
Food Outlets	380
Food Outlet and Advertisement	70

Table 3.1: Children Trajectory Dataset

3.1.2 Food Exposure Model Dataset

We built the Food Exposure model's dataset using a combination of three datasets: Children Trajectory dataset, Neighborhood Mapping dataset, and Egocentric-Food Dataset. First, we used the Children Trajectory dataset, however, we combined the food consumption related classes into one class which is the Food Consumption class. Figure 3.2 shows a sample images for Food Consumption (a), Food Outlets (b), Food Advertisements (c), and Food Outlets and Advertisements (d).

Figure 3.2: Food Exposure sample images for different classes

In addition to the Children Trajectory dataset, we used the Neighborhood Mapping Dataset, which consists of images from the community food environments of the children such as food outlets and food advertisements that are lo-

cated in the neighborhoods of the children's schools[26]. The data was collected using a module that identifies, classifies, and maps food outlets and advertisements within a range of 800-meter around selected schools[26]. The images in this dataset were classified into: Food Outlets, Food Advertisements, or Food Outlets and Advertisements. Table 3.2 shows the number of images per each of the previous classes. In addition, the images in this dataset are very similar to the images in the Children Trajectory dataset since they were taken in the same area. Moreover, the food outlets and advertisements are labeled with subcategories. The food outlets are labeled with different subcategories such as supermarket, grocery shop, butcher shop,etc. The food advertisements are labeled with the name of the advertised food products and if they are healthy or not.

Category	**Count**
Food Outlet	2048
Food Outlet and Advertisement	2130
Food Advertisement	25

Table 3.2: Neighborhood Mapping Dataset

Moreover, we used the EgocentricFood dataset [17] that includes 5,038 food related images that are taken by wearable cameras. From this dataset, we sampled 3000 food consumption images, and 200 food outlet images. Table 3.3 shows the number of images per each class for each dataset. Since the number of images that belong to the Food Advertisement category is very small, we additionally crawled food advertisement images using a major Web search engine (Google),

which are mainly images of food advertisement billboards.

Dataset/Category	Food Consumption	Food Outlet	Food Outlet + Ads	Food Ads
Children Trajectory	2880	380	70	
Neighborhood Mapping		2048	2130	25
EgocentricFood	3000	160		
Crawled Ads				512
Food Exposure	5880	2588	2200	537

Table 3.3: Food Exposure Model Dataset

3.1.3 Food Consumption Model Dataset

We built the Food Consumption model's dataset using food consumption related classes of the Children Trajectory dataset as shown in Table 3.4. Figure 3.3 shows a sample images for Personal Food Consumption (a), Others Food Consumption (b), and Personal and Others Consumption (c).

Category	Count
Personal Food Consumption	1600
Others Food Consumption	340
Personal and Others Food Consumption	940

Table 3.4: Food Consumption Model Dataset

(a) (b) (c)

Figure 3.3: Food Consumption sample images for different classes

3.2 Models

3.2.1 Food Exposure Typology Model

Using the Food Exposure dataset explained above, we trained different multi-label deep learning models that are based on Convolutional Neural Network (CNN) architectures to classify the food exposure images into one or more of the following: Food Consumption, Food Outlets, and Food Advertisements. We split our dataset into 80% for training (8,965 images), 10% for validation (1,120 images) and 10% for testing (1,120 images) in a balanced way between the classes as shown Table 3.3. This data was used to train and test three CNN-based deep

learning models that are used in computer vision tasks which are: MobileNetv1 [27], MobileNetv2 [28] , and VGG16 [29].

The VGG16 is a very deep CNN architecture that is composed of a total of 16 layers where 13 of them are CNN layers followed by 3 fully connected layers. The model contains a total of 134 million parameters. We used a VGG16 model loaded with the pretrained ImageNet weights [30]. We modified the head of the model, which contains the three FC layers, where we replaced the 4096 neurons in the first two FC layers with 500 neurons. Also, we modified the last FC layer, which is the output layer, where we replaced the 1000 neurons that correspond to the number of classes of ImageNet by 3 neurons that correspond to our three classes (Food Consumption, Outlets, and Advertisements). In addition, since our aim is to build a multi-label classification model, we replaced the Softmax activation function with the Sigmoid function that outputs independent probability for each of the classes.

MobileNetV1 is an efficient and accurate CNN model that is compatible with mobile devices. This model is more efficient and more accurate than the VGG16 model discussed above. It is made up of 14 pairs of Convolution (Conv) and Depthwise separable Convolution layers (Conv dw), followed by a Fully Connected layer and a Softmax classifier. The Depthwise Separable Convolutions (DWISE) that was proposed by Chollet [31], is an approach that substitutes the operations that are done in the convolutional layer with a simpler version that breaks the convolution into two separate layers. Each of the Conv and Conv

dw is followed by Batch Normalization and ReLU6 activation function. We used this model loaded with ImageNet weights and we modified the FC layer to be 3 instead of 1000. Also, we used Sigmoid Activation function.

MobileNetV2 is an improved version of MobileNetV1, and it is based on an inverted residual structure with linear bottlenecks. The inverted residual blocks are used to connect the start and the end of a convolutional block by a skip connection, and it has a structure goes in an inverted direction (narrow to wide then to narrow) according to the number of channels [28]. This approach led to a fewer number of parameters (3.4 million). In addition, the linear bottleneck block is composed of Convolution2D layer, Depthwise Separable Convolution layer, and a linear convolution layer with ReLU6 activation function as shown in Figure 3.4. The model is made up of 19 residual bottleneck layers.

In our experiments, we concentrated on MobileNetv2 model since it is one of the top accurate and efficient computer vision pretrained models. First, the size of the model is only 14 MB with 3,538,984 parameters, which make it compatible for deployment in mobile devices and applications which we aim to. The few number of parameters contributes in the model's high inference time of 3.83 ms. Second, the model achieved 0.713 Top-1 Accuracy on ImageNet dataset. Table 3.5 shows the results of the experiments that were done on ImageNet validation dataset for some of the known pretrained models. For example, we can see that InceptionResNetV2 achieved a higher accuracy of 0.803, however, it contains huge number of parameters (55,873,736) compared to MobileNetV2. Therefore,

MobileNetv2 balances between high accuracy and efficiency.

Model	**Size (MB)**	**Top-1 Accuracy**	**Parameters**	**Inference Time (ms)**
MobileNetV2	14	0.713	3,538,984	3.83
InceptionV3	92	0.779	23,851,784	6.86
InceptionResNetV2	215	0.803	55,873,736	10.02
VGG16	528	0.713	138,357,544	4.16

Table 3.5: Pretrained models results on ImageNet validation dataset [1]

Starting from MobileNetv2 architecture loaded also with ImageNet pretrained weights, we created our feature extractor base model by freezing the weights of all MobileNetv2 layers. Then, we added a classifier on the top of our feature extraction model as shown in Figure 3.5. Our classifier is made up of a GlobalAveragePooling2D layer followed by a dense layer of size 256 and a dropout regularization layer (twice). The last layer is a dense layer of size 3 with a sigmoid activation function that outputs independent probabilities. After that, we did fine-tuning by unfreezing the weights of the last 56 layers of the MobileNetV2.

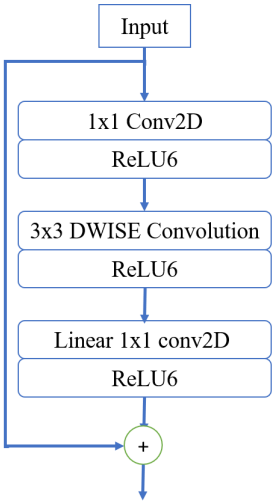

Figure 3.4: MobileNetV2 Bottleneck Block

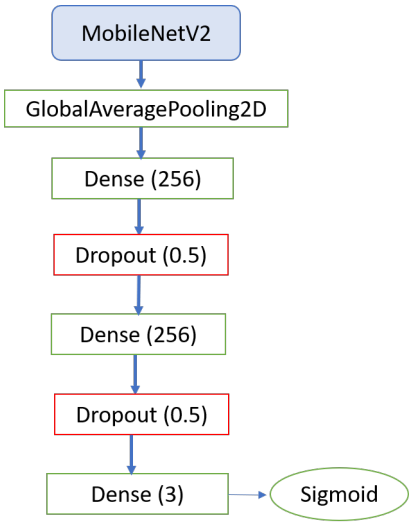

Figure 3.5: The Architecture of our Food Exposure Model

In all of the above models, we tuned the hyper-parameters using the Random Search algorithm [32] by trying the following parameters: batch sizes (32,64,128), Learning rate (0.0001, 0.001, 0.01), and optimizers (Adam, SGD, RMSProp). In addition, we resized the images in all of the experiments to 224×244.

3.2.2 Food Consumption Model

Our multi-label Food Consumption model was trained using the Children Trajectory dataset to classify the images into one or more of the following categories: Personal Food Consumption and Others Food Consumption. We splitted the dataset into 70% for training (2,016 images), 10% for validation (297) and 20% for testing (576) in a balanced way among the classes. As discussed above, MobileNetv2 model is more efficient and more accurate than VGG16 and MobileNetV1 models since it uses an optimized architecture with a fewer number of

parameters. Therefore, we used the MobileNetv2 model that is described in Figure 3.5, however, we changed the output layer to be Dense(2) instead of Dense(3) since we have two main classes in this dataset (Personal Consumption and Others Consumption).

3.3 Results

3.3.1 Food Exposure Model Results

Tables 3.6, 3.7 , and 3.8 show the results of MobileNetV2, MobileNetV1, and VGG16 models respectively on the Food Exposure testing dataset. We evaluated each model using the following metrics: precision, recall, and F1-score. We can clearly notice that MobileNetV2 outperforms the other models in terms all of the metrics. The optimal hyper-parameters that were used in training our MobileNetV2 are the following: For the transfer learning phase, we trained the model for 20 epochs with a batch size of 64 and a learning rate of 0.001. In the fine-tuning phase, the model was trained for 15 more epochs with a learning rate to 0.0001. Also, we used Adam optimizer.

Category/Metric	Precision	Recall	F1-score
Food Consumption	0.99	0.99	0.99
Food Outlet	0.98	0.99	0.98
Food Advertisement	0.95	0.93	0.93
Average	0.97	0.97	0.96

Table 3.6: MobileNetV2 results on the Food Exposure Test Data

Category/Metric	Precision	Recall	F1-score
Food Consumption	0.95	0.94	0.94
Food Outlet	0.94	0.95	0.94
Food Advertisement	0.88	0.90	0.89
Average	0.92	0.93	0.92

Table 3.7: MobileNetV1 results on the Food Exposure test dataset

Category/Metric	Precision	Recall	F1-score
Food Consumption	0.96	0.95	0.95
Food Outlet	0.95	0.94	0.94
Food Advertisement	0.90	0.88	0.89
Average	0.94	0.92	0.92

Table 3.8: VGG16 results results on the Food Exposure test dataset

3.3.2 Food Consumption Model Results

The results of our Food Consumption model on the testing data are shown in Table 3.9. The optimal hyper-parameters that the model was trained with are the following: learning rate 0.001, 25 epochs, Adam optimizer, and a batch size of 32.

Category/Metric	Precision	Recall	F1-score
Personal Food Consumption	0.97	0.99	0.98
Others Food Consumption	0.97	0.89	0.93
Average	0.97	0.94	0.95

Table 3.9: MobileNetV2 results on the Food Consumption Test Data

3.3.3 Error Analysis

In this section, we perfrom error analysis for the Food Exposure and Consumption models. First, the food exposure model performed well on the testing data with

an average f1-score of 0.96. The Food Advertisement class had lower precision and recall than other classes since some of the images contain food advertisements that are very small and barely visible. Second, the food consumption model also performed well on both of its classes. The recall of the others food consumption class was 0.89 since some of the images contain other people consuming food on the same table, however, the wearable camera is directed towards the table and it does not clearly show the other people on the table.

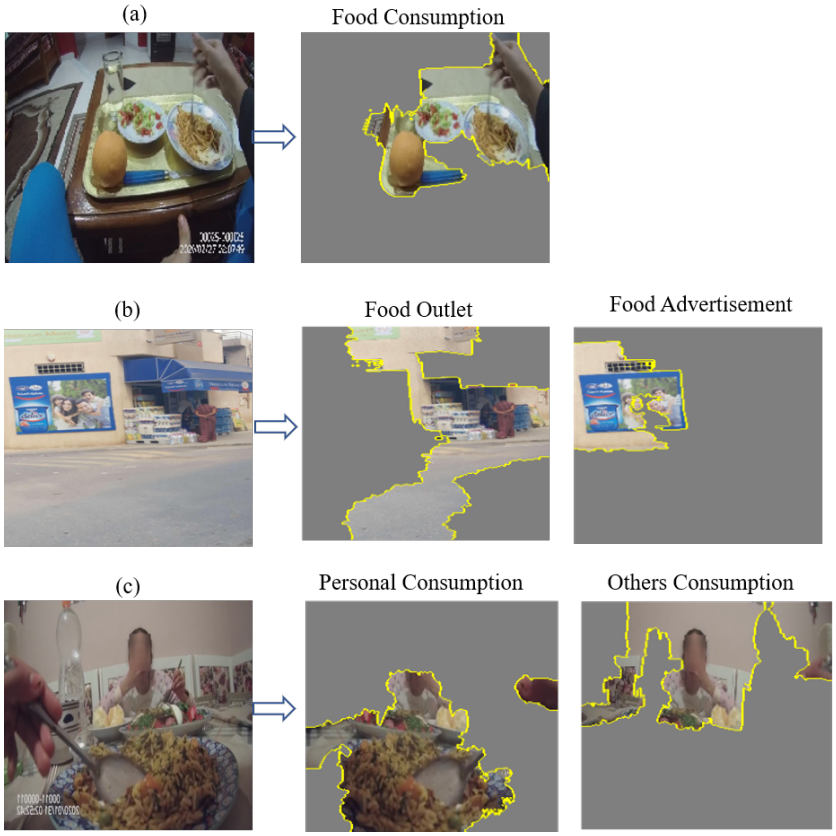

Figure 3.6: LIME results of sample images

In addition, we used LIME [33], which stands for Local Interpretable Model-agnostic Explanation, to explain the two models' decisions on sample images by

extracting the regions that are responsible for the classifiers' predictions. For example, in Figure 3.6, image (b) is classified as Food outlet and advertisement since it contains a region of pixels that belongs to outlets, also it contains another one that belongs to advertisements. Similarly, image (c) is labeled as personal and others food consumption because there is a dish that is directly in front the camera wearer on one side, and on the other side there is another person who is consuming food on the same table.

Chapter 4

Food Item Healthiness Prediction

Automatically estimating the healthiness of food items from food images is a challenging computer vision task . This task is usually composed of two steps which are: (1) detecting and localizing the food items that are present in the images, and (2) building a food item healthiness prediction model. Our proposed approach that is shown in Figure 4.1, takes the Personal Food Consumption related images as an input to our generic food item detection model, which is based on YOLOv3 architecture. Then, the food items are extracted from the images using their corresponding detected bounding boxes. These extracted food items are then used as an input to our NOVA classification model, which is based on the NOVA classification System [9]. This system aims to classify food items into four groups according to the processing level they went through. The four groups are: 1)Unprocessed Foods, 2)Processed Culinary Ingredients, 3) Processed Foods, 4) Ultra-processed Foods.

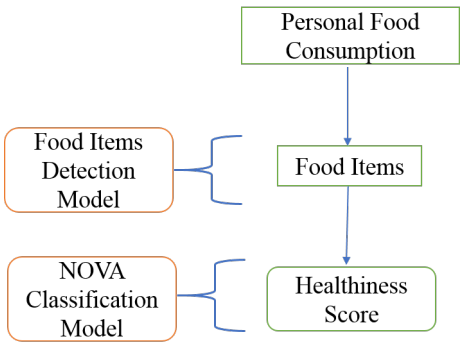

Figure 4.1: Healthiness Prediction Flowchart

4.1 Food Item Detection Model

We developed a generic food item detection model that localizes the food item in an image using bounding boxes. The food items are of various types, shapes, and sizes. Also, they could be in a dish, a bowl, a cup, or held by a person. We trained and tested our food item detection model, which is based on an optimized YOLOv3 architecture, on different datasets.

4.1.1 Datasets

Training our food item detection model requires large datasets of images that contain food items that are annotated using bounding boxes' coordinates. First, we created and labeled a dataset that includes images with various food items using bounding boxes, and we refer to this dataset as the NOVA dataset. In addition, we used two public datasets, which are the EgocentricFood dataset [17] and the UECFood-256 dataset [14]. We assumed that all types of food items in

these datasets belong to one class, which is Food, and we used their bounding box annotations to localize food items in the images. In addition, we converted the bounding boxes coordinates to the normalized YOLOv3 format.

First, we built our NOVA dataset using 1,800 unique images that were sampled from the output of the Food Consumption model. These images belong to either Personal Food Consumption or Personal and Others Food Consumption. In addition, the sampled images contain various types of food items that belong to the different NOVA groups. Second, we created a custom interface on Labelbox platform [25] that allows the annotators to select a food item and label it with the corresponding classes. We trained four Tunisian expert dietitians, since the dataset was captured in Tunisia and it contains food items that are specific to this country, to do the labeling on our Labelbox project and we provided them with the needed instructions. The task was that for each image, the dietitian should select each food item that is present in the image using a bounding box and assign to it one or more of the four NOVA classes. Also, if it is impossible to classify a certain food item into one of the classes, the dietitian can choose either the Unknown Liquid or the Unknown Solid class and specify the reason behind this choice.

To ensure the quality of the labels, each image was labeled twice by two different dietitians and a consensus score that represents the agreement between the labelers was generated. The consensus score was calculated using the IoU of the bounding boxes as well as the chosen classes. After finishing the labelling of

the images, the agreement score between the dietitians was 85%. The discrepancy between the annotations is due to the fact that some food items contain a lot of ingredients and some of them are not visible in the image such as oil and salt. To enhance the quality of the annotations further, we asked the dietitians to go over the labels with a consensus score less than 95% and to agree on the correct labels. After the data was annotated on Labelbox, we ended up with a dataset that contains 4,201 food items that belong to different classes. To train our food items detection model, we assumed all of these food items to belong to only one class, namely Food.

Second, we also used the EgocentricFood dataset, which includes 5,038 images that are taken by wearable cameras. The images contain 7,294 food items that are annotated using bounding boxes and classes. There are nine classes in this dataset which are: glass, cup, jar, can, mug, bottle, dish, food, and basket. The distribution of these classes is shown in Table 4.1. Also, the nature and quality of the images in this dataset are similar to the NOVA dataset that we created since the images were taken in realistic setting using wearable cameras. Figure 4.2 shows a sample of images from this dataset.

Category	Number of Items
Glass	975
Cup	775
Jar	37
Can	176
Mug	1,063
Bottle	2,250
Dish	983
Food	939
Basket	96
Total Number	7,294

Table 4.1: Classes distribution of EgocentricFood dataset

Figure 4.2: Sample images from EgocentricFood dataset

Third, we also used the UECFOOD-256 dataset, which is an Asian food dataset that is crawled from the Web. The dataset is annotated using bounding boxes and classes, which are the names of the food items. It contains a total of 28,898 unique images that include 31,395 food items, which belong to 256 different kinds of food. Each kind of food in the dataset has a 100+ images. Table 4.2 shows the top-10 food items with the highest number of occurrences in the

dataset. Figure 4.3 shows sample images from the UECFOOD-256 dataset. Since a part of the dataset was crawled from the Web, there are images that are not taken in real-life setting such as commercial images of food items.

Category	**Number of Items**
Miso Soup	728
Rice	620
Ramen Noodle	353
Green Salad	342
Beef	246
Hamburger	233
Egg	224
Toast	218
Fried Rice	169
French Fries	153

Table 4.2: Top-10 classes of UECFood-256 dataset

Figure 4.3: Sample images from UECFood-256 dataset

4.1.2 Models

Our generic Food item detection model is based on the architecture of YOLOv3 model with some optimizations applied to it. In the following subsections we will explain: why we chose YOLOv3 model, the model's architecture and output, the optimizations applied to the model, and the experiments that we did on different datasets.

4.1.2.1 YOLOv3 Performance

Our generic food item detection model is based on YOLOv3 (You Only Look Once, Version 3) [8] model that was developed by Redmon and Farhadi. YOLOv3 is a one stage real-time object detection model that localizes specific objects in images and videos using bounding boxes. This model outperforms many object detection models in terms of mean Average Precision (mAP) at IOU threshold of 0.5, as well as the object detection speed.

4.1.2.2 Model's Architecture

YOLOv3 architecture is made up of two main components which are the Feature Extractor and the Feature Detector. The Feature Extractor part is based on Darknet-53, which is a convolutional neural network architecture that is made up of 53 layers, consisting of 3×3 and 1×1 convolutional layers followed by residual connections, proposed by ResNet architecture [34], which are used to take the output from a certain layer and add it into another deeper layer in

the block. In addition to the Darknet-53 architecture, 53 layers are added to serve as a detection head for the network resulting in a total of 106 convolutional layers. The detection head of the model applies the detection on three different image scales through applying 1×1 detection kernels on their corresponding feature maps [8]. The three scales of each image are determined by the Strides parameter in the CNN that are responsible for down-sampling the images by the factors of 32, 16, and 8. We trained our food detection model on images of size 416×416 which results in three resolutions which are: 52×52, 26×26, and 13×13. This technique helps in improving the accuracy of detecting food items of different sizes.

4.1.2.3 Model's Output

After getting the feature maps, the input image is divided into $S \times S$ grid according to the extracted feature map size. For example, a 416×416 image with a 26×26 feature map will result in an image divided into 26×26 cells. Each of the cells predicts 3 bounding boxes, objectness scores, and class predictions. The model outputs bounding box coordinate (x,y,w,h), where (x, y) is the center of the bounding box and (w, h) is the width and height of the box. The bounding boxes are calculated by the help of the anchor boxes which are predefined bounding boxes that are used to predict the bounding boxes coordinates by predicting the offsets to the anchor boxes. Figure 4.4 shows the predicted bounding box coordinates in green and the anchor box in red.

The prior anchor boxes are calculated using k-means [35] which is an unsupervised iterative clustering algorithm that starts by choosing K random points as initial clustering centers (centroids), then calculating the distance from each point to each of the clustering centers, and finally assigning each point to its nearest centroid. During k-means iterations, the centroids are updated until optimal results are reached. In our model, the input of the algorithm is (w, h) of the bounding boxes where w is the width of the bounding box and h is the height of the bounding box. Also, we set $k = 9$ since we need 3 anchor boxes per each of the three image scales. YOLOv3 calculates the distance from the centroid to the box is calculated by subtracting 1 from the IoU of the box and the centroid as shown in Equation 3.1.

$$Distance(Box, Centroid) = 1 - IOU(Box, Centroid) \tag{4.1}$$

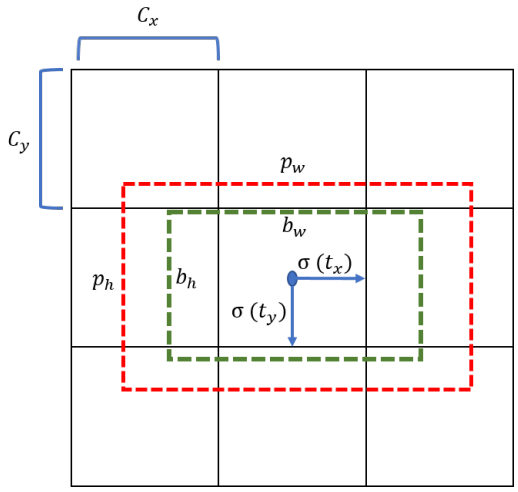

Figure 4.4: Bounding box coordinates prediction

The equations below are used by YOLOv3 model for bounding box coordinates calculation. First, (b_x, b_y, b_w, b_h) is the transformed bounding box coordinates of (t_x, t_y, t_h, t_w) which is the output of the CNN, where (b_x, b_y) is calculated by applying sigmoid function (3.5) on the predicted (t_x, t_y) and adding (c_x, c_y) which is the top-left offset of our grid from the current cell of the feature map. (b_w, b_h) is the width and height of the predicted bounding boxes that are calculated using (p_w, p_h) which is the anchor box's coordinates.

$$b_x = \sigma(t_x) + c_x \tag{4.2}$$

$$b_y = \sigma(t_y) + c_y \tag{4.3}$$

$$b_w = p_w e^{t_w} \tag{4.4}$$

$$b_h = p_h e^{t_h} \tag{4.5}$$

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \tag{4.6}$$

In addition to the bounding box coordinates, the model outputs an objectness score which is calculated using logistic regression and indicates the probability that there is an object is inside a certain bounding box. Moreover, the model predicts classes for the objects using Sigmoid function so that it becomes a multi-label model where it can predicts more than one class per bounding box. YOLOv3 model uses Binary Cross Entropy (BCE) loss for objectness scores and classes predictions.

4.1.2.4 Model Optimization

YOLOv3 model calculates the bounding box error using Mean Squared Error (MSE) of $t - \hat{t}$ [8], where t is the ground truth coordinates, and $\hat{t}$ is the predicted ones. In our model, we used an improved loss function which is Generalized IoU ($GIoU$) proposed by Rezatofighi et al.[36]. First, IoU is a measure that calculates the similarity between two bounding boxes using Jaccard index by dividing the intersection of the shapes by the union of them as shown in Equation 3.2

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{4.7}$$

However, this metric has two weaknesses which are discussed in [36]. First, If $|A \cap B| = 0$, then $IoU = 0$ and therefore IoU doesn't tell us if the bounding boxes are near or far of each others. Second, IoU doesn't actually reflect the overlap between the bounding boxes. $GIoU$ metric was proposed to solve these problems. $GIoU$ is calculated using Equation 4.8 [36], where C is the smallest box enclosing A and B, and $|C \backslash (A \cup B)|$ calculates the area occupied by C without A and B. The IoU value range is $[0, 1]$, however, for $GIoU$ it is $[-1, 1]$, where 1 is the maximum value when two bounding boxes overlap and -1 is the minimum value when the bounding boxes are not overlapping. $GIoU$ loss is calculated by subtracting 1 from the value of $GIoU$ as shown in equation 4.8. Their experiments show that using the $GIoU$ loss improves the AP of many object detection models including

YOLOv3 model (+6.36% AP on COCO dataset [37]).

$$GIoU(A,B) = IoU(A,B) - \frac{|C \backslash (A \cup B)|}{|C|} \tag{4.8}$$

$$\mathcal{L}_{GIoU} = 1 - GIoU \tag{4.9}$$

In addition, instead of using Binary Cross Entropy loss for objectness scores and classes prediction, we used Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss) shown in Equation 3.1. Where y is the true label of the image, $\hat{y}$ is the predicted probability, and σ is the sigmoid function that maps the values between 0 and 1. This is a more stable version of BCE loss since in the case of negative predicted values it will take too long to converge, however when we use BCEWithLogitsLoss it uses sigmoid before applying the BCE loss resulting in a more stable results [38].

$$\begin{aligned} \text{BCEWithLogitsLoss } = -\frac{1}{n} \times \sum_i \left(y_i \times \log\left(\sigma\left(\hat{y}_i\right)\right) + \right. \\ \left. (1 - y_i) \times \log\left(1 - \sigma\left(\hat{y}_i\right)\right)\right) \end{aligned} \tag{4.10}$$

4.1.2.5 Model's Loss Computation

YOLOv3 model optimizes the predicted results using the concept of error back-propagation, where the error is calculated between the predicted value and the real value. Our optimised YOLOv3 model loss function is based on the weighted summation of the food items localization loss, the classification loss and the ob-

jectness loss as shown in Equation 4.11 .

$$\mathcal{L}_{model} = \mathcal{L}_{Localization} + \mathcal{L}_{Classification} + \mathcal{L}_{Objectness} \tag{4.11}$$

$$\mathcal{L}_{Localization} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathcal{L}_{GIoU}(b_i^j, \hat{b}_i^j) \tag{4.12}$$

$$\begin{aligned} \mathcal{L}_{Objectness} = & \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{obj} \left[c_i^j \times \log\left(\sigma\left(\hat{c}_i^j\right)\right) - \left(1 - c_i^j\right) \times \log\left(1 - \sigma\left(\hat{c}_i^j\right)\right) \right] \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{noobj} \left[c_i^j \times \log\left(\sigma\left(\hat{c}_i^j\right)\right) - \left(1 - c_i^j\right) \times \log\left(1 - \sigma\left(\hat{c}_i^j\right)\right) \right] \end{aligned} \tag{4.13}$$

$$\begin{aligned} \mathcal{L}_{Classification} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{obj} \sum_{c \in class} & \left[p(c_i^j) \times \log\left(\sigma\left(p(\hat{c}_i^j)\right)\right) - \left(1 - p(c_i^j)\right) \right. \\ & \left. \times \log\left(1 - \sigma\left(p(\hat{c}_i^j)\right)\right) \right] \end{aligned} \tag{4.14}$$

In the above loss equations, $s^2 = (s \times s)$ is the number of cells of the feature map, and B which is set to 3 in our model is the number of bounding boxes generated by each cell. In the localization loss equation (4.12), b_i^j and $\hat{b}_i^j$ are the true and predicted bounding boxes coordinates respectively. The objectness loss (4.13) is calculated using BCEWithLogitsLoss, where c_i^j and $\hat{c}_i^j$ are the true and predicted confidences respectively. $\mathbb{1}_{i,j}^{noobj}$ is used to determine if the j^{th} bounding box of the i^{th} cell is not responsible for the detection of the object. In addition, λ_{noobj} is the weight of $GIoU$ error which is set to 0.5 in our experiments. Similarly, the classification loss (4.14) is calculated using BCEWithLogitsLoss, where $p(c_i^j)$ is the ground truth probability that the object in the i^{th} cell belongs

to class c, and $p(\hat{c}_i^j)$ is the predicted probability. Also, $\mathbb{1}_{i,j}^{obj}$ checks if the j^{th} bounding box of the i^{th} cell is the one responsible for the detection. It is equal to 1 if the $GIoU(BoundingBox, GroundTruth)$ is the largest, and its is equal to 0 otherwise.

4.1.2.6 Models

After building the architecture of our YOLOv3 model, we trained three models using the UECFood256 dataset, the EgocentricFood dataset, and the Nova dataset.

First, we trained our base food item detection model using our modified YOLOv3 model. Starting from the COCO pretrained weights [37], we trained the model on the UECFood256 dataset. We split the data into 80% for training (23,119 images) and 20% for validation (5,780 images).

After training our base model on the UECFood256 dataset, we froze the weights of our backbone model, Darknet-53, which is used as our feature extractor model in our YOLOv3 architecture. Using the EgocentricFood dataset, we trained the head of our YOLOv3 model using transfer learning. The dataset was split into 80% for training (4,038 images) and 20% for validation (1,000 images).

Finally, our main model is based on the base model that we built using the UECFood256 dataset. We did transfer learning by freezing the backbone of our architecture, Darknet-53, and we trained the head of the model on a combination of the Nova dataset and the EgocentricFood dataset. We split the data into 80%

for training (images) and 20% for testing(images). Also, we trained a standard YOLOv3 model on this dataset and compared it with our modified version of the model.

We tested the above three models on 500 images that were sampled from the Nova dataset, which is described in Section 4.2.1. All the models were trained for 100 epochs on images of size 416×416. Also, we used a learning rate of 0.01 and a decay weight of 0.0005. We used different anchor boxes according to the dataset that was used for training. We calculated 9 anchor boxes for each dataset using the K-mean algorithm that is explained in Section 3.4.2. Table 4.3 shows the anchor boxes for each dataset on three scales.

Dataset/ Scale	**Small (32)**	**Medium (16)**	**Large (8)**
UECFood256	$(195 \times 174), (319 \times 259), (417 \times 369)$	$(535 \times 289), (507 \times 420), (581 \times 381)$	$(445 \times 549), (606 \times 454), (593 \times 574)$
EgocentricFood	$(68 \times 73), (82 \times 124), (182 \times 98)$	$(124 \times 149), (103 \times 233), (225 \times 179)$	$(173 \times 332), (426 \times 193), (402 \times 367)$
EgocentricFood+Nova	$(63 \times 54), (76 \times 104), (158 \times 87)$	$(98 \times 152), (107 \times 247), (177 \times 172)$	$(339 \times 151), (180 \times 320), (411 \times 303)$

Table 4.3: The calculated anchor boxes for each dataset

4.1.3 Results

We evaluated our models using the mean Average Precision (mAP) metric. mAP is the average of AP over the classes and since we only have the Food class, the mAP will be the same as AP. To calculated this metric, we need to compute the precision (4.15) and the recall (4.16). Also, we need to specify an IoU threshold value. For threshold $IoU = 0.5$, the model classifies the object detection as True Positive (TP) if $IoU \geq 0.5$, else it will consider it as wrong detection and classifies

it as False Positive (FP). In case there is a food item in the image and the model couldn't detect it, we classify it as False Negative (FN). Finally, we classify all the parts of the image that don't contain any food item as True Negative (TN).

$$Precision = \frac{TP}{TP + FP} \tag{4.15}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.16}$$

In our experiments, we used two mAP evaluation metrics: the Pascal Visual Object Classes (VOC) metric [39] and the COCO metric [37]. The Pascal VOC metric, mAp@0.5 calculates the mAP for $IoU \geq 0.5$. The COCO metric, mAP@[0.5 : 0.95] , calculates the average of mAP for different IoU thresholds that range from 0.5 to 0.95 with a step size of 0.05.

The results of the three models on their corresponding validation datasets are shown in they below Table 4.4. We can notice from the results that our YOLOv3 model achieved good mAP@0.5 scores on the validation sets of each model.

Model	**Precision**	**Recall**	**mAP@0.5**	**mAP@[0.5:0.95]**
UECFood256 (base model)	0.90	0.93	0.91	0.65
EgocentricFood	0.87	0.91	0.90	0.66
NOVA and EgocentricFood	0.86	0.89	0.88	0.63

Table 4.4: The Results of the models on the validation datasets

Table 4.5 shows the results on the validation set of the NOVA and EgocentricFood combined dataset on the standard YOLOv3 model and our modified

version of the model. Our modified YOLOv3 model outperformed the standard YOLOv3 in terms of mAP@0.5 (3% improvement) as well as it achieved a lower localization and objectness losses. This is due to using $\mathcal{L}_{GIoU}$ instead of MSE for calculating the error between the ground truth and the predicted bounding boxes.

Model	**Localization Loss**	**Objectness Loss**	**mAP@0.5**
Yolov3	0.12	0.08	0.85
Yolov3+GIoU	0.01	0.02	0.88

Table 4.5: YOLOv3 vs.YOLOv3+GIoU results on the validation dataset

In order to compare the models, we tested the above three models on 500 annotated images that were sampled from our NOVA dataset. Table 4.6 presents the results using different metrics. We can clearly see that the NOVA and EgocentricFood model outperforms the first two models. UECFood256 model achieved bad results since the images in this dataset are crawled from the Web and they are homogeneous and preprocessed, which is different from the nature of the images in our dataset that are captured from wearable cameras. The EgocentricFood model achieved better results since the images in this dataset are taken from wearable camera which are very similar to the images in our dataset. Finally, combining the NOVA and the EgocentricFood datasets and training the head of our YOLOv3 with frozen weights of Darknet-53 taken from the UECFood256 model achieved the best results over all of the used metrics as shown in the below table.

Model	Precision	Recall	mAP@[0.5:0.95]	mAP@0.5
UECFood256	0.44	0.22	0.11	0.22
EgocentricFood	0.70	0.73	0.41	0.71
NOVA and EgocentricFood	0.87	0.91	0.65	0.90

Table 4.6: The results of the models on the NOVA dataset

4.1.4 Error Analysis

In this section, we analyze the error of the food item detection model. As discussed in the previous section, the NOVA and EgocentricFood model that is based on the optimized version of YOLOv3 outperformed all other models. We can see from table 4.6 that the model achieved a high mAp@0.5 of 0.90, as well a precision of 0.87 and a recall of 0.91 on the testing dataset. The model was able to detect most of the food items that appear in the Food Consumption images as shown in Figure 4.5. On the other hand, the model was not able to detect some of the food items that are occluded by other objects such as the dish in image (a).1, which is occluded by the water bottle. On the other hand, overall the model was robust, as it was able to detect small food items that are far from the food table (see image (b).1). This is due to the fact that our YOLOv3 model applies the detection on three different image scales (small, medium, and large). This factor decreased the precision, by increasing the false positives, since the labelers did not label food items that are far away from the food table. Finally, some of the food items have overlapping bounding boxes(image (b).2), we solved this by excluding the ones with a low confidence score (lower than 0.40).

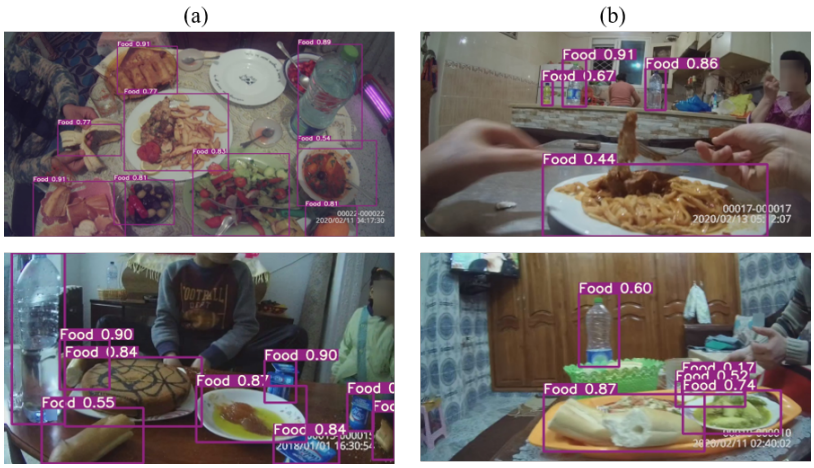

Figure 4.5: Sample results of the food item object detection model

4.2 NOVA classification Model

There have been many attempts to estimate the healthiness of food items using their corresponding images. However, most of these are quantitative approaches that are based on volume and calories estimation, which face many limitations including capturing the images from specific predefined angles, adding reference objects in each image, and segmenting the food items in a pixel-wise way. Thus, many nutrition experts are advocating for food classification systems that are based on the food processing level rather than using calories and nutrients content of the food items [40]. Therefore, our approach aims to do a qualitative assessment of the healthiness of food items rather than a quantitative assessment that is based on the amount of calories of the food items.

In this thesis, we used the NOVA classification system [9] that classifies food

items into four groups according to their nature, extent and the aim of the industrial processes that they were applied to the food items. The first group is the Unprocessed foods, which includes natural foods such as vegetables, fruits, eggs, milk, water, etc. The second group is the Processed culinary ingredients that are usually acquired from group 1 and they include butter, oil, honey, etc. The third group is the Processed foods that are included in meals and dishes such as ham, fish, meat, bread, etc. Finally, we have the Ultra-processed foods group that includes foods that are produced using a series of industrial processes such as chips, chocolate, soft drinks, hotdog, etc. These four groups are related to each other since a food item may belong to more than one group at the same time.

To reach our goal, we built a dataset of food items that are classified according to the NOVA classification system described above. This data was used to train and test our deep learning multi-label model that classifies each food item into one or more of the NOVA system groups.

4.2.1 Datasets

Using the NOVA dataset that was discussed previously, we extracted the food items from the images with their corresponding labels using the bounding boxes coordinates. The initial dataset consisted of 4,016 images that belong to the different NOVA groups. Also, it contained 112 images that were labeled with the

Unknown Liquid category and 73 images that were labeled with the Unknown Solid category. We excluded those categories since they do not belong to any of the NOVA groups. In addition to that, we cleaned the dataset by removing the images that are too blurry or very small. The NOVA dataset, after cleaning it, consisted of 3,728 images that belong to the different NOVA groups as shown in table 4.7

NOVA Groups	**1**	**2**	**3**	**4**	**12**	**13**	**14**	**23**	**34**	**123**	**124**	**134**
Images Nb	1,094	122	759	557	510	40	4	7	11	569	49	6

Table 4.7: NOVA Dataset Groups

4.2.2 Models

After building the NOVA dataset using the procedure described in the previous section, we split it into 80% for training, 10% for validation and 10% for testing. This dataset was used to train and test our multi-label classification model that aims to classify each food item into one or more of the NOVA groups.

For this task, we used MobileNetV2 architecture. First, we built a model that is made up of MobileNetV2 architecture loaded with frozen ImageNet weights, and we added a classifier on the top of it. The classifier consisted of a global average pooling layer followed by a dense layer of 250 neurons and a drop out layer of value 0.5. The output layer is a dense layer with 4 neurons representing the NOVA groups. Also, we used sigmoid activation function to output independent probabilities for the classes we have. We trained this model for 20 epochs with

Adam optimizer that has a learning rate of 0.001. After that, we did a fine-tuning step by unfreezing the last 55 layers of MobileNetV2 model and retrain the model for 10 more epochs with a learning rate of 0.0001.

Since the NOVA dataset contains images of various sizes, we used different image sizes as hyper-parameter (128, 160, 192, 224). We resized the images using Bilinear Interpolation, which is a resampling method that calculates a new pixel value based on the distance weighted average of the nearest four pixels [41].

4.2.3 Results

After training our NOVA classification model with different image sizes, the 224 × 224 image size was the best fit for the model. The results of this model on the testing data is shown in Table 4.8 that presents the precision, the recall, and the F1-score for each of the NOVA groups.

Group/Metric	**Precision**	**Recall**	**F1-score**
1	0.92	0.86	0.89
2	0.90	0.85	0.87
3	0.86	0.84	0.85
4	0.84	0.85	0.84

Table 4.8: NOVA model results on the testing data

4.2.4 Error Analysis

In this section, we analyze the error of our NOVA classification model. The overall performance of the model on the testing data was relatively high, however, some of the images were miss-classified due to the complexity of the food items. We

noticed that our model could not predict all the ground truth NOVA groups for some of the images that contain ingredients that are not visible to the model such as salt and oil. Figure 4.6 shows a sample of miss-classified images by the NOVA model. For example, the first image's ground truth NOVA groups are 1-2 since it is a salad, and 4 because it contains cheese. The model was able to correctly predict that the food item belong to groups 1 and 2, however, it didn't predict the group 4 since most of the salad related images in the training dataset belong to groups 1 and 2 only. Another example is image 3, where the ground truth is that the food item belongs to the NOVA groups 1, 3, and 4 since it contains bread, tomatoes, and cheese. The model correctly predicted that it belongs to groups 1 and 3 and it misses group 4. These results explain why we don't have a very high recall in some of the NOVA groups.

This NOVA classification model is used to qualitatively asses the healthiness of the Tunisian food specifically, since the model was trained on images that were collected in Tunis where there are some type of foods that are specific for this country. From here, we can conclude that our NOVA classification model has two limitations 1) Dependent on the country where the dataset is collected 2) Doesn't always detect the small and hidden ingredients.

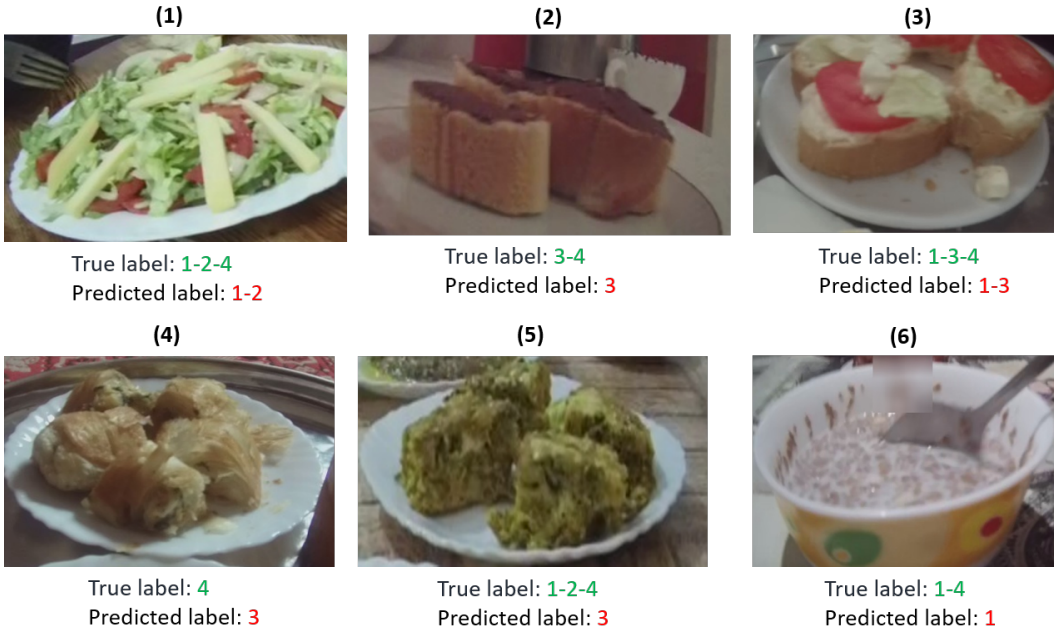

Figure 4.6: Error analysis of miss-classified images of the NOVA model

4.3 Overall System Performance

In this section, we will evaluate the performance of the whole approach.

First, we sampled 100 images from our testing datasets that belong to the different classes of Food exposure. Then, we ran our models sequentially on the sampled images and we calculated the corresponding precision, recall, and F1-score. Table 4.3 shows the results of each model, as well as the final average score of the approach. our approach achieved an average f1-score of 0.85. The reason behind this score is the propagation of error between the models. For example, if the Food consumption model miss-classifies an image , it will affect the results of the food item detection model, as well as the NOVA classification model.

Model	**Precision**	**Recall**	**F1-score**
Food Exposure	0.97	0.95	0.96
Food Consumption	0.92	0.93	0.92
Food Item Detection	0.79	0.80	0.79
NOVA Classification	0.76	0.74	0.75
Average	0.86	0.85	0.85

Table 4.9: Overall performance of our approach

Chapter 5

Case Study

5.1 Setting

Our case study was done in Greater Tunis on a sample that included 215 children aged 11-12 years old from grades 5 and 6 recruited from 29 Tunisian schools between January 30th until mid-March 2020. After filtering the collected data by the wearable cameras through a binary image classification model that classify images into food and non-food classes, we ended up with a total of 30,170 images (3.69 GB) related to food exposure [24]. In this thesis, we applied our approach that is composed of four deep learning models on the resultant dataset.

5.2 Results

First, we applied the Food exposure and the Food Consumption models on the dataset. We found that the vast majority of pictures were showing food consumption (95%) with only 4% related to food outlets. Among the food consumption images, 69% were about Personal Food Consumption" (69%), 23% were images including both personal and other people consumption of foods, and only 8% were images showing only other people consuming foods. After that, we subjected the images that were labeled with Personal Food Consumption to the food item healthiness prediction models. First, we ran the food item detection model on the images, and we extracted the food items using bounding box coordinates (27,757 food item). Then, we applied the NOVA classification model on those extracted food images. Table 5.1 shows the number of images per the NOVA group(s). These results will be used by researchers to analyze the eating behaviours of students and associate them with the food exposures that they encountered including food advertisements and food outlets.

NOVA Groups	**1**	**2**	**3**	**4**	**12**	**13**	**14**	**23**	**34**	**123**	**124**	**Other**
Images Nb	8,101	764	6,062	2,441	3,342	179	5	14	33	4,909	58	1,849

Table 5.1: NOVA results of the Tunisian case study

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we presented a novel approach that is based on various deep learning models that classifies food exposure related images in the wild into a hierarchy of food exposure and consumption classes, and extracts and assesses the healthiness of the food items based on the NOVA classification system. This approach can be used by researchers to automate the dietary case studies that are usually done using traditional tools such as interviews and questionnaires, and which usually suffer from low accuracy and recall. Our approach is based on a series of four deep learning models: Food Exposure Model, Food Consumption Model, Food Item Detection Model, and Food Item Healthiness Prediction Model.

First, we built a multi-label Food Exposure model that aims to classify the images into one or more of the following classes: Food Consumption, Food Outlet,

and Food Advertisement. Then the multi-label Food Consumption model comes after, and it is used to classify the Food Consumption images into: Personal Food Consumption, Others Food Consumption, or both of them. To train these model, we built two high quality datasets of images that belong to the above-mentioned classes for each model using crowdsourcing on Labelbox platform.

Second, we built the food item healthiness prediction models that consists of the Food Item Detection model followed by the NOVA Classification model. The Food Item Detection model is based on a modified version of YOLOv3 model, where we used $GIoU$ loss instead the MSE for the localization error. This optimization improved the precision and the recall of detecting food items. The model was trained using a new approach where we used a combination of three datasets, UECFOOD256 dataset, the Egocentric dataset, and the NOVA dataset that we built, to train the model on one class which is Food. Our NOVA dataset was labeled by Tunisian expert dietitians where each food item was labeled by bounding box coordinates with its corresponding NOVA group. Finally, we trained the multi-label NOVA classification model which aims to qualitatively asses the healthiness of the detected food items according to their processing level using the four NOVA groups.

We deployed these models in a real-world case study in Tunisia, where researchers were able to have for each student the captured images classified into the different food exposure categories. In addition, the extracted food items with their NOVA healthiness categories helped researchers to associate between

the eating behaviours of the children and the food exposures that includes food advertisements and outlets.

Since the models were trained on images that were captured in a specific country, Tunisia, they may not result in a good accuracy when applied directly on images that are captured from different countries. First, countries have different buildings infrastructure which results in a different food outlets and advertisements forms. Second, each country has type of foods that are specific to their culture where they are not consumed by people in Other countries. This puts limitations on using the NOVA classification model on Food items from different countries.

6.2 Future Work

For the future work, we are planning to integrate the models we built in this thesis into a user friendly End-to-End tool that can be used by researchers who are working on similar case studies in the Arab region. To reach that, we first need to adapt our models to the new datasets that will be collected in other countries. This could be done by using transfer learning techniques on our trained models or by retraining the models on the combined datasets which makes the models more generalizable. Finally, the approach that we implemented in this thesis will be used in a similar case study in Lebanon.

Appendix A

Abbreviations

CNN	Convolutional Neural Network
ML	Machine Learning
AI	Artificial Intelligence
RNN	Recurrent Neural Network
NLP	Natural Language Processing
GAP	Global Average Pooling
IoU	Intersection over Union
mIoU	mean Intersection over Union
GIoU	Generalized Intersection over Union
YOLOv3	You Only Look Once version 3

Appendix B

Labelbox Crowdsourcing Guidelines

B.1 Food Exposure Labeling Guidelines

The aim of this job is to classify an image related to food exposure into one or more of four categories. These images were captured through wearable cameras, and the classification will help us analyze the data captured. The four categories are: Food Outlets, Self Food Consumption, Others Food Consumption, and Food Advertisements. You will be displayed an image and your task is to decide whether the picture belongs to one or more of the above four categories. For each category, you will have to choose whether the image belongs to it (Yes), doesn't belong to it (No). If it's not clear or you can't decide, you should choose "Not sure".

- An image should be labeled as Food Outlet if it contains a food outlet such as a supermarket, a shop, a vending machine, a restaurant, a mobile vendor, a food-stand, a kiosk, a food factory, etc. Hotels and malls which usually include food outlets are also included in this category.

- An image should be labeled as Self Food Consumption if it contains food items that the person wearing the wearable camera is obviously consuming or is about to consume. For instance, dishes on a table where the person wearing the camera is sitting on, or a sandwich held by the person wearing the camera, etc.

- An image should be labeled as Others Food Consumption if it contains other people consuming food or about to consume food.

- An image should be labeled as Food Advertisement if it contains any food ads such as billboards, storefront ads, commercials on a TV screen, commercials on vehicles, ads in magazines, ads on our mobile phone, etc.

- Finally, if an image doesn't satisfy any of the criteria above, you should label all the categories with "No".

- Note that the displayed image could be labeled using more than one of the labels above if it satisfies more than one criteria. Also note that by food here we mean both food and beverages.

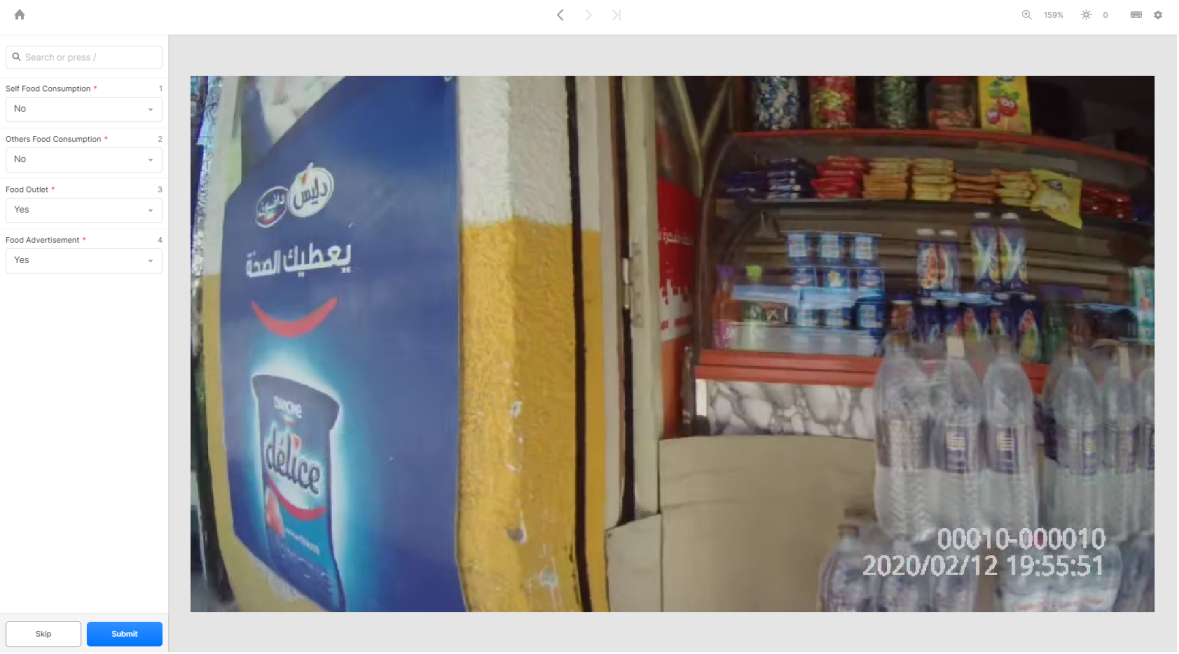

Figure B.1: Food Exposure sample labeled image in Labelbox

B.2 Nova Healthiness Score Labeling Guidelines

The Nova classification assigns food items to four groups according to the extent and purpose of the industrial processing used. For each image, you will have to select each food item that appears in the image and assign one or more group to it.

I. An image will appear to you on the platform. For each food item that appears in the image, you should do the following:

1. You click on the Bounding Box label that is found on the left side of the screen to draw a bounding box on the food item.

2. After drawing the bounding box, a checklist of 6 options will appear on the

left side.

3. If you know the group(s) that the food item belongs to, you should choose one or more of the following 4 groups:

    a) Group 1= Unprocessed or minimally processed foods

    b) Group 2= Processed ingredients

    c) Group 3= Processed foods

    d) Group 4= Ultra-processed foods

4. If you can't choose the group(s) that the food item belongs to, you should choose one of the following options:

a) Unknown- Solid: After choosing this option, you should write the reason in the text box named "Unknown solid reason" which will appear on the left side.

b) Unknown-Liquid: After choosing this option, you should write the reason in the text box named "Unknown liquid reason" which will appear on the left side.

II. Once you are done with the image, you should click on the Submit button to go to the next image.

Figure B.2: NOVA sample labeled image in Labelbox

Bibliography

[1] K. Team, "Keras documentation: Keras applications."

[2] J. Fuhrman, "The hidden dangers of fast and processed food," *American Journal of Lifestyle Medicine*, vol. 12, no. 5, pp. 375–381, 2018.

[3] S. Scaglioni, V. De Cosmi, V. Ciappolino, F. Parazzini, P. Brambilla, and C. Agostoni, "Factors influencing children's eating behaviours," *Nutrients*, vol. 10, p. 706, 05 2018.

[4] A. Wilson, A. Magarey, and N. Mastersson, "Reliability of questionnaires to assess the healthy eating and activity environment of a child's home and school," *Journal of obesity*, vol. 2013, p. 720368, 07 2013.

[5] C. Boushey, D. Kerr, J. Wright, K. Lutes, D. Ebert, and E. Delp, "Use of technology in children's dietary assessment," *European journal of clinical nutrition*, vol. 63 Suppl 1, pp. S50–7, 03 2009.

[6] M. Ranaweera and Q. H. Mahmoud, "Virtual to real-world transfer learning: A systematic review," *Electronics*, vol. 10, no. 12, 2021.

[7] R. Alazaidah and F. K. Ahmad, "Trending challenges in multi label classification," *International Journal of Advanced Computer Science and Applications*, vol. 7, 10 2016.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.

[9] C. A. Monteiro, G. Cannon, M. Lawrence, M. L. da Costa Louzada, and P. P. Machado, "Ultra-processed foods, diet quality, and health using the nova classification system," *FAO*, 2019.

[10] A. Akhi, F. Akter, T. Khatun, M. Uddin, Mohammad, and S. Uddin, "Recognition and classification of fast food images recognition and classification of fast food images," *Global Journal of Computer Science and Technology*, vol. 18, 07 2018.

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

[12] 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," 06 2016.

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

[14] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014.

[15] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[16] E. Aguilar, B. Remeseiro, M. Bolaños, and P. Radeva, "Grab, pay, and eat: Semantic food detection for smart restaurants," *IEEE Transactions on Multimedia*, vol. 20, pp. 3266–3275, 2018.

[17] M. Bolaños and P. Radeva, "Simultaneous food localization and recognition," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3140–3145, 2016.

[18] Y. Liang and J. Li, "Deep learning-based food calorie estimation method in dietary assessment," *ArXiv*, vol. abs/1706.04062, 2017.

[19] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2calories: Towards an automated mobile vision food diary," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1233–1241, 2015.

[20] Y. Lu, T. Stathopoulou, M. F. Vasiloglou, S. Christodoulidis, Z. Stanga, and S. Mougiakakou, "An artificial intelligence-based system to assess nutrient intake for hospitalised patients," *IEEE Transactions on Multimedia*, vol. 23, pp. 1136–1147, 2021.

[21] J. Gao, W. Tan, L. Ma, Y. Wang, and W. Tang, "Musefood: Multi-sensor-based food volume estimation on smartphones," 03 2019.

[22] K. Sudo, K. Murasaki, T. Kinebuchi, S. Kimura, and K. Waki, "Machine learning–based screening of healthy meals from image analysis: System development and pilot study," *JMIR Formative Research*, vol. 4, p. e18507, 10 2020.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, 2017.

[24] Z. J. Y, "Ai system in the real world: Capture children food exposure using wearable cameras 2020," 2020.

[25] "labelbox: the leading training data platform for data labeling," 2021.

[26] A. C., "School neighborhood food environment and schoolchildren's diets and nutrition in a middle-income arab city," 2022.

[27] B. Khasoggi, E. Ermatita, and S. Samsuryadi, "Efficient mobilenet architecture as image recognition on mobile and embedded devices," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, p. 389, 10 2019.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[31] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017.

[32] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.

[33] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, p. 18, 12 2020.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 06 2016.

[35] Y. Li and H. Wu, "A clustering method based on k-means algorithm," *Physics Procedia*, vol. 25, pp. 1104–1109, 12 2012.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," 02 2019.

[37] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[38] Pytorch, "Bcewithlogitsloss," 2019.

[39] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.

[40] M. Crino, B. T, H. Trevena, and N. B., "Systematic review and comparison of classification frameworks describing the degree of food processing," *Nutrition and Food Technology: Open Access*, vol. 3, 01 2017.

[41] G. R. Arce, J. Bacca, and J. L. Paredes, "3.2 - nonlinear filtering for image analysis and enhancement," in *Handbook of Image and Video Processing (Second Edition)* (A. BOVIK, ed.), Communications, Networking and Multimedia, pp. 109–IV, Burlington: Academic Press, second edition ed., 2005.