Food Analytical Methods (2022) 15:3045–3078
https://doi.org/10.1007/s12161-022-02353-9

Allergen30: Detecting Food Items with Possible Allergens Using Deep Learning-Based Computer Vision

Mayank Mishra[1] · Tanmay Sarkar[2] · Tanupriya Choudhury[1] · Nikunj Bansal[1] · Slim Smaoui[3] · Maksim Rebezov[4,5] · Mohammad Ali Shariati[6] · Jose Manuel Lorenzo[7,8]

Received: 20 June 2022 / Accepted: 26 June 2022 / Published online: 5 July 2022

Abstract

Food allergies impose a significant health concern on the community. A small number of certain food items can cause an allergic reaction within the human body. The symptoms can range from mild hives or itchiness to life-threatening anaphylaxis. In most cases, such reactions can be prevented by simply being aware of the allergen-based food items and avoiding the consumption of the same. We are among the first research attempts to train a deep learning–based object detection model to detect the presence of such food items within an image. We introduce our Allergen30 dataset, which hosts more than 6,000 annotated images of 30 commonly used food items that can trigger an adverse reaction. We report the comparison of multiple variants of the current state-of-art object detection methods, YOLOv5 and YOLOR. Furthermore, we qualitatively analyzed the performance of these methods by surveying the predictions made on the test dataset images.

Keywords Food allergy · Food dataset · Deep learning · Computer vision · Object detection

Introduction

A food allergic reaction is an abnormal response by an individual's body toward specific food items that are otherwise commonly tolerated by a more significant proportion of

Mayank Mishra, Tanmay Sarkar, Tanupriya Choudhury and Nikunj Bansal contributed equally to this work.

✉ Tanmay Sarkar
tanmays468@gmail.com

✉ Jose Manuel Lorenzo
jmlorenzo@ceteca.net

Mayank Mishra
mmayank74567@gmail.com

Tanupriya Choudhury
tanupriya1986@gmail.com

Nikunj Bansal
nikunj8126@gmail.com

Slim Smaoui
slim.smaoui@cbs.rnrt.tn

Maksim Rebezov
rebezov@yandex.ru

Mohammad Ali Shariati
shariatymohammadali@gmail.com

[1] Informatics Cluster, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand 248007, India

[2] Department of Food Processing Technology, Government of West Bengal, Malda Polytechnic, Bengal State Council of Technical Education, Malda 732102, West Bengal , India

[3] Laboratory of Microorganisms and Biomolecules of the Center of Biotechnology of Sfax, 3018 Sfax, Tunisia

[4] Department of Scientific Research, V. M. Gorbatov Federal Research Center for Food Systems, 26 Talalikhin st, Moscow 109316, Russian Federation

[5] Biophotonics Center, Prokhorov General Physics Institute of the Russian Academy of Science, 38 Vavilov st, Moscow 119991, Russian Federation

[6] Department of Scientific Research, K. G. Razumovsky Moscow State University of Technologies and management (The First Cossack University), 73 Zemlyanoy Val, Moscow 109004, Russian Federation

[7] Centro Tecnológico de La Carne de Galicia, Parque Tecnológico de Galicia, Avd. Galicia nº 4, San Cibrao das Viñas, 32900 Ourense, Spain

[8] Área de Tecnología de los Alimentos, Facultad de Ciencias de Ourense, Universidade de Vigo, 32004 Ourense, Spain

people. This intolerance toward particular foods is individualistic; a large majority remains unaffected by consuming the same food items (Taylor and Hefle 2001). Commonly, certain proteins that naturally occur in foods trigger an abnormal immune response within the human body (Bush and Hefle 1996; López-Pedrouso et al. 2020). One particular category based on the nature of such an immune response is the immediate hypersensitivity reaction. In such reactions, intolerance symptoms begin to manifest within a few minutes to an hour after consuming a reaction-causing food item. The responses are marked by the emergence of allergen-specific immunoglobulin E (IgE) antibodies (Mekori 1996). These allergic reactions that are IgE-mediated are widespread and extremely dangerous (Ortolani and Pastorello 2006). Food allergies are capable of causing harm to the respiratory tract, cardiovascular system, gastrointestinal tract, and skin and can produce life-threatening anaphylactic shocks to people of all ages (Nowak-Węgrzyn et al. 2017). Gastrointestinal hypersensitivity may lead to excessive diarrhea, vomiting, nausea, and abdominal pain within 2 h of eating the offending food (Abrams and Sicherer 2016). Symptoms of respiratory disturbances like asthma and rhinitis and cutaneous symptoms like eczema, urticaria, angioedema, and pruritus are among the most common manifestations of food allergies (Taylor and Hefle 2001).

The occurrence of reactions due to oral intake of food items is highly common among people. A study conducted in the USA (Gupta et al. 2019) shows that about 20% of adults from the USA suffer from some kind of food allergy, and nearly 38% of them have at least once visited the hospital's emergency department due to a food allergic reaction in their lifetime. Moreover, multiple pieces of research from the last 20 years show that up to 20% of the population from the leading industrialized countries have suffered from mild to severe adverse reactions after eating a particular food (Pereira et al. 2005; Nwaru et al. 2014). However, almost 90% of the food products that cause IgE-mediated allergic reactions come from a particular set: milk, egg, peanut, and wheat (Bousquet et al. 1998). The items from this specific set are also known as the "priority food allergens," and any food product that contains these as ingredients can also cause an allergic reaction (Boye 2012). Based on the presence of allergen compounds, most intolerant foods can be classified as lactose-, histamine-, caffeine-, gluten-, or salicylate-rich foods, or ovomucoid-rich food. Milk and milk-based products are high in lactose. Due to casein in the solid part and whey protein in the liquid part of the milk, it can cause intolerance in consumers. Products of microbial fermentation like aged cheese, processed meat, and alcohol are rich in histamine and can induce high discomfort in people who cannot metabolize ingested histamine (Maintz and Novak 2007; Comas-Basté et al. 2020). A vast share of the population showcases intolerance toward products high in caffeine, including coffee, tea, and chocolates. Wheat is an essential source of carbohydrates and high in protein. It can be classified as a gluten product and can cause a severe allergic reaction within 10–60 min of consumption (Muthukumar et al. 2020). Food items like oranges, currants, and apples have increased salicylate content, which can lead to respiratory tract discomfort in many people (Baenkler 2008). Egg-based food items have high content of ovomucoid, which can cause severe allergic reactions. Even though the effect of food allergens on people is widespread, there is no standard treatment or cure for such reactions (Muthukumar et al. 2020). An effective remedial method is to take precautionary measures by eliminating allergic food from the diet altogether. However, in many cases, people suffer from intolerance due to their inability to detect food items with possible allergens.

Deep learning is a subset of artificial intelligence that uses deep computational models with multiple layers of abstraction to analyze patterns and intricate structures in data (LeCun et al. 2015). With the introduction of deep convolutional nets, breakthroughs have been achieved in the domain of image, video, speech, and audio processing (Hinton et al. 2012; Tompson et al. 2014; Krizhevsky et al. 2017). In our work, we exploit the recent advancements in the area of object detection in deep learning to build a system to analyze an input image and detect the presence of any food item with possible allergens. Our research will allow people to be more informed about the food they consume and will aid in preventing a possible life-threatening allergic reaction.

Literature Review

Instead of relying on prior hard-coded information, machine learning (ML) gives the ability to a system to make decisions by analyzing patterns from raw data. However, the performance of many ML algorithms is highly dependent on the representation of the data. These algorithms can generate a correlation between the given set of features and the output, but they do not influence how the features are defined. This poses a substantial challenge; it is almost impossible to pre-define the perfect set of features for the algorithms to learn from complex problems. Deep learning addresses this problem by using deep computational models to construct high-level abstract features by building on low-level features. Table 1 summarizes the historical trends in deep learning. Deep learning–based image recognition methods are spread across various industries. They are being used to detect diseases in the medical industry, for pest control in the agriculture sector, to defect detection in the manufacturing industry, and to build autonomous machines in the transportation sector (Sun et al. 2018a, b; Islam et al. 2019; Jia et al. 2019).

Table 1 Summarizing the historical trends in deep learning

Wave of development	Time frame	Main idea/result	Reference
Cybernetics	(1940s–1960s)	Developed the McCulloh-Pitts neuron, one of the first models representing a brain function. Weights to this linear model were set manually, and the model could map a set of inputs to two outputs	McCulloch and Pitts (1943)
		Proposed the perceptron, the first model that was able to learn weights using the given inputs	Rosenblatt (1958)
		Their model, adaptive linear element or ADALINE, updated the weights using a training algorithm, a particular version of the currently used stochastic gradient descent	Widrow and Hoff (1988)
Connectionism	(1980s–1990s)	Introduced distributed representation where input is defined as features, and each feature represents many possible inputs	Hinton et al. (1986)
		Used backpropagation for neural networks by repeatedly updating weights to minimize the difference between the actual output and the calculated output	Rumelhart et al. (1986)
		Built a digit recognition system using backpropagation and achieved a 1% error rate on the US Postal Service zip code digits	LeCun et al. (1989)
		Solved mathematical difficulties in modeling long sequences with a neural network by introducing the long short-term memory (LSTM)	Hochreiter and Schmidhuber (1997)
Deep learning	(2000s–present)	Efficiently trained a neural network called deep belief using a greedy layer-wise pretraining strategy	Hinton et al. (2006)
		Won the ILSVRC-2012 competition using a deep convolutional neural network (DCNN) by achieving a top 5 error rate of 15.3%	Krizhevsky et al. (2017)
		Proposed a DCNN called Inception that won the ILSVRC-2014 using a 22 layered architecture with a constant computational budget	Szegedy et al. (2015)
		Used a 16–19 layered CNN with 3 × 3 filters and 1 × 1 stride and pad to secure the first in the localization and the second places in the classification tracks of ILSVRC-2014	Simonyan and Zisserman (2014)
		Introduced a residual learning framework to train very deep CNNs. Deployed a CNN with 152 layers that won the ILSVRC-2015 classification, detection, and localization tasks	He et al. (2016)
		Built an end-end model architecture for object detection that determined the presence of objects from the regions proposed by the network	Ren et al. (2015)
		Performed fast real-time object detection by making predictions over a limited number of bounding boxes instead of using proposed regions	Redmon et al. (2016)
		Improved efficiency of object detection by using convolutional neural network's pyramidal feature hierarchy	Liu et al. (2016b)
		Put forward the RetinaNet, a one-stage dense object detector that increased object detection performance using featurized image pyramid and focal loss	Lin et al. (2020)

Such a widespread application in almost every industry can be attributed to the deep learning methods being accurate, efficient, and consistent (Liu et al. 2018a, b; GC et al. 2021). Similarly, architectures like convolutional neural networks (CNNs) are commonly used to recognize food categories and to perform food *vs*. non-food classification (Zaidi et al. 2022; Singla et al. 2016). Table 2 tabulates the use of deep learning for image analysis in the food industry.

Various food datasets have been created to train computer vision models. Table 3 shows a description of several such datasets including the number of images and food categories they contain. Table 4 summarizes the research undertaken in food classification. Kawano and Yanai (2014) created the UECFood-100 Dataset (available at http://foodcam.mobi/dataset100.html), which contains 100 Japanese food categories to implement a food recognition system. It consists of 14,361 images

Table 2 Application of deep learning in food image analysis

Aim of research	Model	Dataset	Accuracy	Reference
Development of food recognition system (FRS)	ConvNet	Food-101 (https://www.kaggle.com/kmader/food41)	99.14%	Jahani Heravi et al. (2018)
Detection of adulteration in red-meat	CNN	31 fat, 18 lamb, 13 pork samples	94.4%	Al-Sarayreh et al. (2018)
Classification of *Chrysanthemum* varieties	DCNN, SVM,	7 varieties; 11,038 images	100%	Wu et al. (2018)
FRS development	DCNN + Inception V3	Food-101 (https://www.kaggle.com/kmader/food41) UEC Food 100 (http://foodcam.mobi/dataset100.html) UEC Food 256 (http://foodcam.mobi/dataset256.html)	82.46%	Zheng et al. (2018)
Dietary assessment	CNN-stacked sparse autoencoder (SAE)	UEC Food 100 (http://foodcam.mobi/dataset100.html) UEC Food 256 (http://foodcam.mobi/dataset256.html)	94.6%	Liu et al. (2018a)
FRS for Chinese food	ResNet	Food-101 (https://www.kaggle.com/kmader/food41)	94.1%	Fu et al. (2017a)
Classification of food	ResNet-50	Food-475 (http://www.ivl.disco.unimib.it/activities/food475db/)	95.79%	Ciocca et al. (2018)
FRS development	DNN	Food-101 (https://www.kaggle.com/kmader/food41)	95.45%	Martinel et al. (2018)
FRS development	DCNN	UEC Food 100 (http://foodcam.mobi/dataset100.html) UEC Food 256 (http://foodcam.mobi/dataset256.html)	95.15%	Yanai and Kawano (2015)
Viable spore count and concentration estimation of *Clostridium sporogenes*	CNN	hyperspectral images of *Clostridium sporogenes* inoculated plates	90–94%	Soni et al. (2021)
Quality assessment of nuts	CNN	2300 images of 289 samples	95.83%	Han et al. (2021)
Identification of acrylamide in potato chips	DCNN, MobileNetv2	4560 images	98.24%	Arora et al. (2021)
Assurance of seed quality	DCNN	1920 seeds	96.9–100%	Zhang et al. (2021)
Peach leave disease detection	CNN	Peach leaves (https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw/color)	98.75%	Yadav et al. (2021)
Non-destructive quality assessment	YOLOv3-Lite	–		Hu et al. (2021)
Counting of kernels and yield estimation of corn	VGG 16	19,848 images	RMSE = 60.27 and MAE = 41.36	Khaki et al. (2021)
Grape maturity classification	CNN	–	Syrah = 93.41%, Cabernet Sauvignon = 72.66%	Ramos et al. (2021)
Beef cut classification	VGG 16	1113 images of different beef cuts	98.6%	GC et al. (2021)

Table 2 (continued)

Aim of research	Model	Dataset	Accuracy	Reference
Predictive model for lead concentration determination	WT-SAE, DNN	1120 lettuce leaf	Coefficient of determination = 0.959	Zhou et al. (2020)
Fish freshness determination	VGG 16	48 fish samples	98.21%	Taheri-Garavand et al. (2020)
Egg sorting system	VGG 16	315 images of 105 eggs	94.84%	Nasiri et al. (2020)
Apple defect detection	CNN	300 apples	96.5%	Fan et al. (2020)
Tomato defect detection	ResNet50	43,843 images	97.70%	da Costa et al. (2020)
FRS development	CNN	Food-101	58.65%	Tatsuma and Aono (2016)
FRS development	AlexNet	UEC-Food 100	78.77%	Yanai and Kawano (2015)
FRS development	CNN	UNIMIB2016 (http://www.ivl.disco.unimib.it/activities/food-recognition/)	79.00%	Ciocca et al. (2017a)
FRS development	CNN	THF Food 50 (https://paperswithcode.com/dataset/thfood-50)	92.30%	Termritthikun et al. (2017)
FRS development	NutriNet	225,953 images (different drink and food)	94.47%	Mezgec and Koroušić Seljak (2017)
FRS development	Ensemble Net	Food-101	96.61%	Pandey et al. (2017)
Calorie estimation	Multi-task CNN	Different recipe webpages (http://park.ajinomoto.co.jp/; https://www.kikkoman.co.jp/homecook/; http://allrecipes.com/; http://chainer.org/)	Relative error = 41.70%	EGE and YANAI (2018)
Calorie estimation	GoogLeNet CNN	Food 201 (https://github.com/janarez/Food201) Restaurant (https://www.kaggle.com/tags/restaurants)	76.00%	Myers et al. (2015)
Plum classification based on varieties	AlexNet	525 images of Angelino, Black-Splendor, and Owent varieties	91–97%	Rodríguez et al. (2018)
Defect detection in mangosteen	CNN	120 images	97.50%	Azizah et al. (2017)
Diseased and pest affected apple detection	CNN	4000 images	97.50%	Tan et al. (2016)
Blueberry damage detection	ResNet, ResNeXt	737 blueberries	88.44%	Wang et al. (2018)
Classification of normally or artificially ripened banana	AlexNet	240 images	90.00%	B.S. et al. (2018)
Classification of three peach disease	Deep belief network	270 samples	82.5–100%	Sun et al. (2018b)
Soluble solid content and firmness prediction of pear	SAE-FNN	180 samples	For soluble solid content and firmness, the coefficient of determination are 0.91 and 0.89, respectively	Yu et al. (2018a)
FRS development	VGG 19, VGG 16, ResNet	25,000 images	92.00%	Zhang et al. (2018)
Shrimp freshness detection	SAE	256 samples	96.55%	Yu et al. (2018b)

Table 2 (continued)

Aim of research	Model	Dataset	Accuracy	Reference
Predictive model for total volatile nitrogen content in shrimp	SAE, SVM	240 samples	Coefficient of determination = 0.921	Yu et al. (2019)
Credit evaluation system in Food supply chain	DNN	–	90.00%	Mao et al. (2018)
Prediction of morbidity for disease due to contaminated food infection in the gastrointestinal tract	DNN	119 types of food 227 types of contaminants	Success rate = 58.50%	Song et al. (2017)
Identification of beetles that are potential food contaminators	ANN, SVM	6900 images 15 species of beetles	80–85%	Bisgin et al. (2018)
FRS development	DCNN	Chinese menu (https://github.com/AkishinoShiame/Chinese-Menu-Recognition-App-Dataset)	82%	Lee et al. (2017)
Nutritional density assessment	DNN	390 samples	92.2%	Pfisterer et al. (2018)
Food safety modeling	Deep radial basis function NN	Food poisoning data in China from 2013–2015	–	Geng et al. (2019)
Personalized diet recommendation system	CNN, RNN	100 billion words (Google News)	72.8%	Chen et al. (2018)
Transformation of food category	Conditional CycleGAN	Food change lens https://negi111111.github.io/FoodChangeLensProjectHP/	–	Naritomi et al. (2018)
Fake food recognition	NutriNet	520 beverages and foods	92.18%	Mezgec and Seljak (2019)
Personalized diet management	SENet, ResNet, DenseNet, ResNeXt,	Imagenet (https://image-net.org/)	95.7%	Sahoo et al. (2019)
Foreign object detection in walnut	CNN	781 images	99.50%	Rong et al. (2019)
FRS development for electronic sales	CNN, Tree Adaptation Network	Meal 300, Office 31 (https://domain-adaptation.wixsite.com/tan-meal300)	76.27–100%	Xiao et al. (2020)
Detection of pesticide in apple	AlexNet-CNN	48 samples	99.09%	Jiang et al. (2019)
Waste potato identification	VGG	100,314 images	83.3–99.79%	Jagtap et al. (2019)

Table 3 Description of famous food datasets

Dataset	Number of images and categories	Description	Dataset Link	Reference
Food-50	5000 food images spread across 50 kinds of food categories	Crawled food images from the web and selected 100 relevant images manually for each food category	https://github.com/chakkritte/THFOOD-50	Termritthikun et al. (2017)
UECFood-100	14,361 images across 100 food categories containing popular Japanese dishes	Includes bounding box coordinates that point to the location of the food items in every image	http://foodcam.mobi/dataset100.html	Kawano and Yanai (2014)
UNICT-FD889	3583 images of 889 distinct plate of food	Captured images of the same dish multiple times to introduce geometric variability (from various angles) and photometric effect (with and without flashlight) by using a smartphone	https://iplab.dmi.unict.it/UNICT-FD889/	Farinella et al. (2014)
Food-101	101 food categories with 1000 images for each category	Downloaded data from an online collection of user-uploaded food images Compiled diverse but visually and semantically similar food classes including but not limited to waffles, onion rings, pad Thai, pizza, and chocolate cakes	https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/	Bossard et al. (2014)
UECFood-256	256 food categories containing 25,088 images of popular Japanese dishes	Improved upon the UECFood-100 dataset	http://foodcam.mobi/dataset256.html	Kawano and Yanai (2015)
UPMCFood-101	101 food categories with 790–956 images for each category	Category list created using ETHZFood-101 dataset and images collected using a google search engine Added the word "recipes" to each label in the query to decrease the noise in searching results	https://www.kaggle.com/gianmarco96/upmcfood101 http://visiir.lip6.fr/explore	Gallo et al. (2020)
UNICT-FD1200	Consist of overall 4754 images of 1200 distinct plate of food	Improved upon the UNICT-FD889 dataset	https://iplab.dmi.unict.it/UNICT-FD1200/	Farinella et al. (2016)
Instagram 800 k	808,000 food-related images	Crawled images with associated metadata from Instagram over a 6-week period in 2014–2015	https://homepages.inf.ed.ac.uk/thospeda/downloads.html	Rich et al. (2016)
Food-11	16,643 food images grouped in 11 major food categories	Contains a mix of images from other food datasets	https://www.epfl.ch/labs/mmspg/downloads/food-image-datasets/	Singla et al. (2016)
Food-5 K	2500 food images and 2500 images of other items	Dataset is a mix of images from other food datasets, social media platforms, and self-clicked images Dataset used for food/non-food classification and food recognition	https://www.epfl.ch/labs/mmspg/downloads/food-image-datasets/	Singla et al. (2016)
VIREOFood-172	Consist of 110,241 food images spread across 172 food categories	Compiled the dataset from "Go cooking" and "Meishijie" websites by removing duplicate images Used Baidu and Google image search engine for crawling images	http://vireo.cs.cityu.edu.hk/VireoFood172/	Chen and Ngo (2016)

Table 3 (continued)

Dataset	Number of images and categories	Description	Dataset Link	Reference
Food524DB	247,636 food images belonging to 524 food categories	Created by combining the multiple databases including Food-50, Food-101, UEC FOOD-256, and VIREO Food-172	http://www.ivl.disco.unimib.it/activities/food524db/#:~:text=Food524DB%20is%20the%20largest%20publicly,belonging%20to%20524%20food%20categories	Ciocca et al. (2017b)
ChineseFoodNet	180,000 food images belonging to 208 categories	Collected images not only from web recipes and menu pictures but also from real dishes, recipe, and menu	https://sites.google.com/view/chinesefoodnet	Chen et al. (2017)
FoodX-211	Consist of 158,000 images spread across 211 food categories	Used web image search for downloading images for each food class	https://github.com/karansikka1/iFood_2019	Kaur et al. (2019)
ISIA Food-500	399,726 food images spread across 500 food categories	Category list created using Wikipedia and image collected using multiple search engines	http://123.57.42.89/FoodComputing-Dataset/ISIA-Food500.html	Min et al. (2020)
FFoCat	58,962 images belonging to 156 food categories	Contains food categories of the Mediterranean diet	https://zenodo.org/record/5840047#.YnkMoOhBxPY	(Donadello and Dragoni 2019)
Food 2 K	1,036,564 images belonging to 2,000 food categories	Highly diverse image collection with larger volume and category coverage than the previous datasets	http://123.57.42.89/FoodProject.html	Min et al. (2021)

containing popular Japanese dishes. Liu et al. (2016a, b) reported a top 1% accuracy of 76.3% and top 5% accuracy of 94.6% using the DeepFood Model fine-tuned on GoogleNet architecture. Moreover, Fu et al. (2017a, b) reported a top 1% accuracy of 80.6% and top 5% accuracy of 95.9%. They use a logistic regression model with pre-trained ResNet model features. In addition, Zheng et al. (2018) were able to achieve an accuracy of 86.5% using Inception-v3 with Food Part CNN, and Martinel et al. (2018) registered a top 1% accuracy of 89.6% and top 5% accuracy of 99.2% by using CNN-based structure called WISeR with a slice convolution unit to extract vertical food characteristics followed by deep residual blocks.

The Food-101 Dataset contains 101 food categories and 1000 pictures for each food category. With their DeepFood model fine-tuned on GoogleNet architecture, Liu et al. (2016a, b) secured a top 1% accuracy of 77.4% and top 5% accuracy of 93.7%. Moreover, Pandey et al. (2017) built the FoodNet architecture, which is an ensemble-based CNN architecture incorporating fine-tuned GoogleNet, ResNet, and AlexNet. Their model gains a top 1% accuracy of 72.1% and top 5% accuracy of 91.6%. In addition, Fu et al. (2017a, b) were able to achieve a top 1% accuracy of 78.5% and top 5% accuracy of 94.1% by using a logistic regression model with pre-trained ResNet model features. Using an Inception-v3 with Food Part CNN, Zheng et al. (2018) secured an accuracy of 88.0%. Furthermore, Martinel et al. (2018) CNN-based WISer reports a top 1% accuracy of 90.3% and top 5% accuracy of 98.7%.

As an expansion to the UECFood-100 Dataset, Kawano and Yanai (2015) modeled the UECFood-256 dataset (available at http://foodcam.mobi/dataset256.html), which contains 25,088 images of popular Japanese dishes spread across 256 food categories. Liu et al. (2016a, b) DeepFood model secured a top 1% accuracy of 54.7% and top 5% accuracy of 81.5% on the UEC-Food-100 dataset. On the other hand, the logistic regression model of Fu et al. (2017a, b) was able to improve on the DeepFood model by claiming top 1% accuracy of 71.2% and top 5% accuracy of 91.1%. The Food Part CNN of Zheng et al. (2018) reported an accuracy of 78.6%, which was surpassed by McAllister (2018) with a top 1% accuracy of 83.1% and top 5% accuracy of 95.4%.

To classify food and other items, the balanced binary classification Food-5 K Dataset (available at https://www.epfl.ch/labs/mmspg/downloads/food-image-datasets/) is a popular choice (Singla et al. 2016). It contains a subset of pictures from UECFood-100, UECFood-256, and Food-101 Dataset, along with 2500 pictures of non-food items. Singla et al. (2016) showed the efficacy of a fine-tuned GoogleNet Model by achieving 99.2%

Table 4 Summarizing research undertaken in food classification

Model	Dataset	Top 1% accuracy	Top 5% accuracy	Model description	Reference
GoogleNet (fine-tuned)	Food-5 K	99.2	–	Fine-tuned Google Net architecture	Singla et al. (2016)
	Food-11	83.6	–		
DeepFood	UECFood-100	76.3	94.6	Fine-tuned Google Net architecture	Liu et al. (2016a)
	UECFood-256	54.7	81.5		
	Food 101	77.4	93.7		
FoodNet	Food 101	72.1	91.6	CNN based ensemble network architecture incorporating fine-tuned AlexNet, GoogLeNet, and ResNet	Pandey et al. (2017)
ResNet	UECFood-100	80.6	95.9	Feature extraction using a pre-trained ResNet with a logistic regression model for final classification	Fu et al. (2017b)
	UECFood-256	71.2	91.1		
	Food 101	78.5	94.1		
Inception module	UECFood-100	76.3	94.6	Fine-tuned GoogleNet architecture	Liu et al. (2018a)
	UECFood-256	63.6	87		
	Food 101	77	94		
Inception-v3 + FP-CNN	UECFood-100	86.5	–	Food-part CNN framework (FP-CNN) framework for mid-level feature extraction from food images complemented by a fine-tuned Inceptionv3	Zheng et al. (2018)
	UECFood-256	78.6	–		
	Food 101	88.0	–		
Wide-slice residual networks (WISeR)	UECFood-100	89.6	99.2	CNN based structure called WISeR with a slice convolution unit to extract vertical food characteristics followed by deep residual blocks	Martinel et al. (2018)
	UECFood-256	83.1	95.4		
	Food 101	90.3	98.7		
ResNet-152 + ANN	Food-5 K	98.8	–	ResNet-152 is a deep residual pre-trained CNN	McAllister et al. (2018)
	Food-11	91.34	–		
ResNet-152 + SVM (RBF)	Food-11	89.99	–	SVM RBF Kernel is used with ResNet-152 Model	McAllister et al. (2018)
ResNet-152 + SVM (Poly)	Food-11	88.86	–	SVM Polynomial Kernel is used with ResNet-152 Model	McAllister et al. (2018)
SGLANet	ISIA Food-500	63.4	88.9	Stacked Global–Local Attention Network (SGLANet) that jointly learns global and local features	Min et al. (2020)
	Food 101	90.9	98.3		
	Vireo Food-172	91.0	98.3		
PRENet	Food 2 K	83.7	97.3	Fusion of global features and local features learnt and enhanced via the progressive region Enhancement network (PRENet) to recognize more detailed features	Min et al. (2021)
Ensemble (ResNeXt-101, DenseNet-161)	UECFood-100	90.02	–	Ensemble method with the bagging strategy over DenseNet-161 and ResNeXt-101	Arslan et al. (2022)

accuracy, whereas McAllister (2018) reported an accuracy of 98.8% using the ResNet-152 model. The Food-11 Dataset (available at https://www.epfl.ch/labs/mmspg/downloads/food-image-datasets/) recognizes 11 major food categories: dessert, bread, eggs, meat, dairy products, fried food, noodles/pasta, rice, seafood, soup, and vegetable/fruit (Singla et al. 2016). On the other hand, McAllister (2018) achieved an accuracy of 91.34% using their deep neural network trained with features extracted from the ResNet-152 model. They also reported accuracy of 89.99% using SVM with radial basis function kernel and 88.86% using SVM with polynomial kernel.

Food Intolerances

Food intolerances can cause an adverse reaction within the human body. In this section, we discuss typical food intolerances prevalent in the community. Table 5 summarizes the symptoms and food items of specific food intolerances.

Table 5 Symptoms and food items about common food intolerances

Intolerance	Symptoms	Primary food items
Lactose	Abdominal pain/cramps, bloating, gas, diarrhea	Milk and milk-based products
Histamine	Itching of eyes and nose, nasal congestion, vomiting, nausea, diarrhea	Alcoholic beverages, avocado, tomatoes, spinach, eggplant
Gluten	Breathing difficulty, chronic diarrhea, nausea	Wheat, barley, rye products, white bread, pasta, roti
Salicylate	Nasal congestion, asthma, diarrhea, gut inflammation	Pineapple, spinach, eggplant, strawberry
Caffeine	Tremors, anxiety, insomnia	Coffee, tea, energy drinks, soft drinks
Ovomucoid	Skin inflammation, nausea, chest tightness	Egg-based food items

Lactose Intolerance

Lactose intolerance is caused due to the inability of the body to digest lactose. This can be because of an inadequate amount of enzyme lactase in the body which helps to metabolize lactose. Lactose is one of the primary sources of energy for an infant; however, it seems to lack any significant nutritional value for adults. Over time, the amount of lactase in the human body decreases after weaning. This enzyme is located in the small intestine and is responsible to split and hydrolyze the dietary lactose. The deficiency of the lactase enzyme causes lactose malabsorption in the small intestine, which further leads to gastrointestinal problems, including abdominal pain, bloating, and diarrhea (Szilagyi and Ishayek 2018). Milk and food items prepared using milk are natural sources of lactose. The degree of malabsorption of lactose varies among individuals; nevertheless, people with lactose intolerance must be cautious while consuming dairy products. Although in less quantity, lactose is also added to other food items including but not limited to bread and baked goods, candies, breakfast cereals, etc. (Szilagyi and Ishayek 2018).

Histamine Intolerance

Similar to lactose intolerance, histamine intolerance is also caused by an enzyme deficiency. Histamine plays a vital role in the proper functioning of the body—it helps to keep organs working by contracting the smooth muscle tissue of the lungs, uterus, and stomach; assists in dilating blood vessels; and stimulates gastric acid secretion in the stomach. However, if the body lacks adequate amounts of diamine oxidase, the enzyme essential to break the histamine, it starts to accumulate in the body. This can lead to several undesirable symptoms like itchy skin, red eyes, headaches, and swelling of the face. Extreme intolerance can also cause heart palpitations, and reduced blood pressure. Fermented foods and drinks contain increased levels of histamine. Therefore, people with histamine intolerance should be wary of alcoholic beverages, vinegar, aged cheese, and meat. Other food items like avocados, tomato, eggplant, and spinach can also increase the levels of histamine in the body (Comas-Basté et al. 2020).

Gluten Intolerance

Gluten is a type of protein that is seen in grains like wheat, barley, and rye. Gluten sensitivity can be attributed to three main disorders—autoimmune celiac disease (CD), wheat allergy, and non-celiac gluten sensitivity (NCGS). Wheat allergy can be caused due to the proteins present in the wheat, including gluten. Symptoms like nausea, rash, nasal congestion, and breathing difficulty may be seen usually right after consuming wheat and can be life-threatening in extreme situations. In the case of CD, the immune system reacts abnormally to the ingestion of gluten. It can hamper the ability of the small intestine to absorb nutrients and can also lead to permanent intestinal damage. The symptoms of CD include chronic diarrhea, stomach ache, and vomiting and can cause health problems like weight loss, short stature, and delayed puberty. NCGS refers to the gluten sensitivity in people with no CD or wheat allergy. The most common symptoms are mental fatigue (brain fog), bloating, and headache. One must avoid food items prepared from gluten-containing grains (wheat, barley, rye). Many baked goods like cookies, pastries, and white bread also contain gluten. Gluten helps to improve the quality of dough and, therefore, can be traced in pasta, dumplings, and roti (Sergi et al. 2021).

Salicylate Intolerance

Salicylate is a part of a group of chemicals derived from salicylic acid and is difficult to be metabolized by people with salicylate intolerance. Even consuming a small amount of salicylate can trigger adverse reactions in the body. It is believed that the overproduction of leukotrienes in the body causes sensitivity toward salicylate. Respiratory and intestinal tracts are affected causing symptoms like nasal congestion, asthma, diarrhea, and gut inflammation. Salicylates

are found in fruits (pineapple, blueberry, strawberry, blackberry), vegetables (eggplant, broccoli, spinach), and spices (ginger, turmeric, oregano) (Baenkler 2008).

Caffeine intolerance

Caffeine is naturally found in substances like coffee beans, cocoa beans, and tea leaves. It is one of the most common stimulants in the world which affects the nervous system. Adenosine is a neurotransmitter whose levels rise gradually when the brain is active. High levels of adenosine make one feel tired, signaling the body to take a rest. Caffeine connects to the adenosine receptors and does not activate them, making one feel alert and active. Many individuals are sensitive to caffeine, causing unwanted symptoms like faster heart rate, anxiety, jitters, restlessness, and insomnia. High amounts of caffeine are expected in beverages like coffee, espresso, tea, and energy drinks (Temple et al. 2017).

Ovomucoid Intolerance

Ovomucoid is an allergen primarily found in egg whites. The thick and thin layers of egg whites are due to varying quantities of ovomucin content (Stadelman 2003). People, especially children, experience an allergic reaction when their immune system overreacts to the proteins in eggs by considering them as unwanted foreign invaders. The symptoms of ovomucoid intolerance include indigestion, diarrhea, problems in breathing, and swelling of the tongue or lips. In an adverse condition, egg allergy can lead to anaphylaxis, causing the body to go in shock. People who suffer from ovomucoid intolerance are advised to avoid consuming any food items made from eggs.

Object Detection and Deep Learning

Object detection is a common problem in computer vision that focuses on identifying and locating an object belonging to a particular class (e.g., human, cars, chair) from digital videos or images. Some prominent application of object detection includes counting of vehicles; face detection; and detection of road signs, pedestrian, cars, etc., for self-driving vehicles. The problem of object detection dates back to the advent of computer vision. It is an essential component of image understanding and several methods were proposed to tackle this problem even before the neural network–based deep learning rose to the scene. Out of the many traditional methods for object detection, Haar cascades (Viola and Jones 2001) and HOG + Linear SVM (Dalal and Triggs 2005) stand out because of their decent detection performance and speed. Both these methods share two important subcomponents:

(a) *Sliding windows*—a window that slides over the image from left to right and top to bottom, making classification at every region of interest along the way. This window helps to detect the exact location of an object within the image.
(b) *Image pyramid*—an input image is sequentially reduced in dimensions in such a way, that if they were stacked together, they would represent a pyramid. In each of these images, the sliding window tries to detect the presence of an object. The images of decreasing scale within the pyramid assist the sliding window to detect the object of varying sizes (depending on the distance of the object from the camera).

After applying these components, there remains a high chance that the method reports multiple detections of a single object in the image. Non-maxima suppression (NMS) is applied in such an event to keep one detection with the

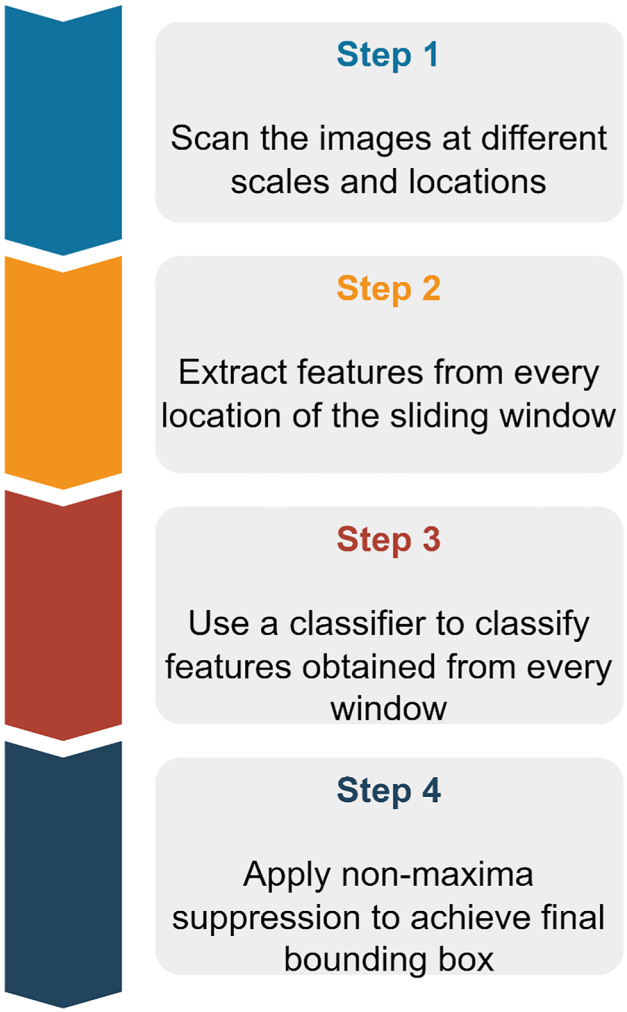


Fig. 1 Fundamental steps in traditional object detection

highest confidence. These fundamental steps of traditional object detection are summarized in Fig. 1.

However, our primary focus in this work is the deep learning–based object detection methods. Deep learning is a subset of machine learning based mainly on the artificial neural network. These networks are inspired by the functioning of neurons within the human body. A neural network consists of multiple layers of "neuron" units, where each neuron is responsible for performing a simple arithmetic operation. Unlike the traditional paradigm, deep learning allows a model to learn recognizable patterns by training on a large set of inputs.

A Basic Neural Network

Figure 2 illustrates the basic computation of a neural network. The network inputs are x_1, x_2, … x_n that are fed into the network as a feature vector. This vector is usually the raw pixel intensities of an image for computer vision tasks like classification or object detection. A bias of constant 1 is added to the input. These inputs are connected to the next neuron. A weight vector W contains the weight value w_1, w_2, … w_n associated with each input value.

The weighted sum of the input vector is performed, following which an activation function is applied to the sum to see if the neuron is activated or not. In the case of deep learning, some popularly known activation functions are the sigmoid, tanh, ReLU, and LeakyReLU. The output of this simple neural network can be mathematically written as:

$$f\left(\sum_{i=1}^{n} w_i x_i\right) \tag{1}$$

where f is the activation function.

The most common and vastly used neural network architecture is the feedforward neural network. The output from the nodes of one layer is fed into the nodes of the next layer. This architecture contains only forward connections; i.e., the output from the nodes of one layer is fed as an input to the nodes of the subsequent layer, thus earning the name feedforward. Figure 2 contains a sample of a feedforward neural network. Layer 0 is the input layer, which contains the vector fed into the network. Layer 1 and layer 2 are the hidden layers that hold 2 and 3 nodes each. The computation between the inputs (or the output of any layer) and the node in the subsequent layer follows the basic computation discussed in the previous section. At last, layer 3, known as the output layer, gives the final computation of the network. Usually, the number of nodes in this layer equals the class labels in the computer vision task at hand. For example, the output layer of the network used to classify handwritten numbers will contain 10 nodes, where each node will convey the potential output corresponding to each label (i.e., digits 0 to 9). According to the requirement of a problem, the number of hidden layers and nodes within a layer can vary in various feedforward neural networks.

A feedforward neural network can be understood as a network where each layer performs a function f. The network, therefore, is a representation with multiple functions composed together. In our example case of Fig. 3, the neural network performs a function $f(x)$ which is

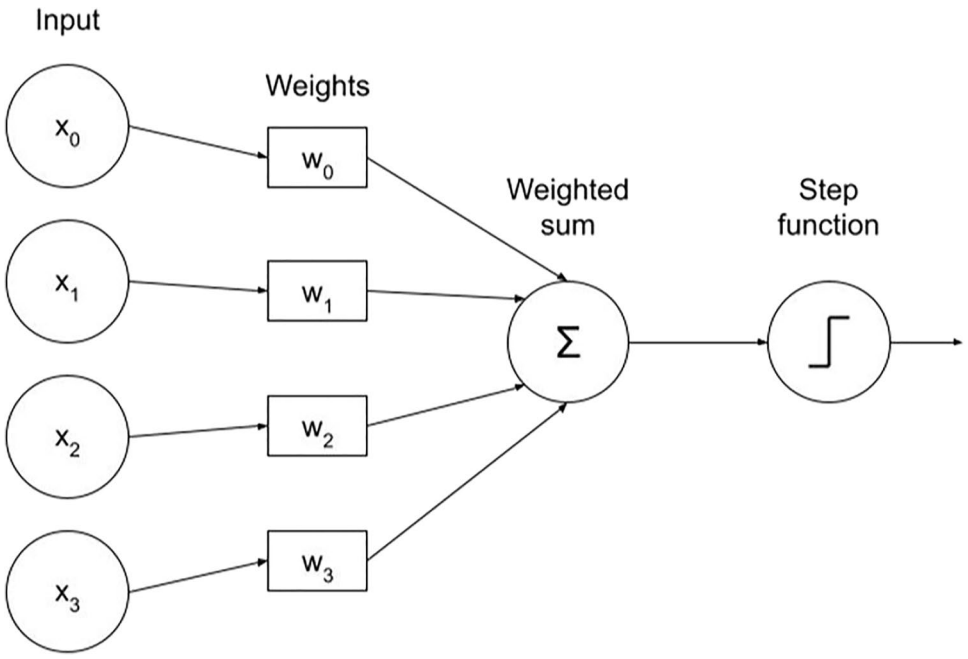


Fig. 2 Understanding a perceptron that makes up a neural network

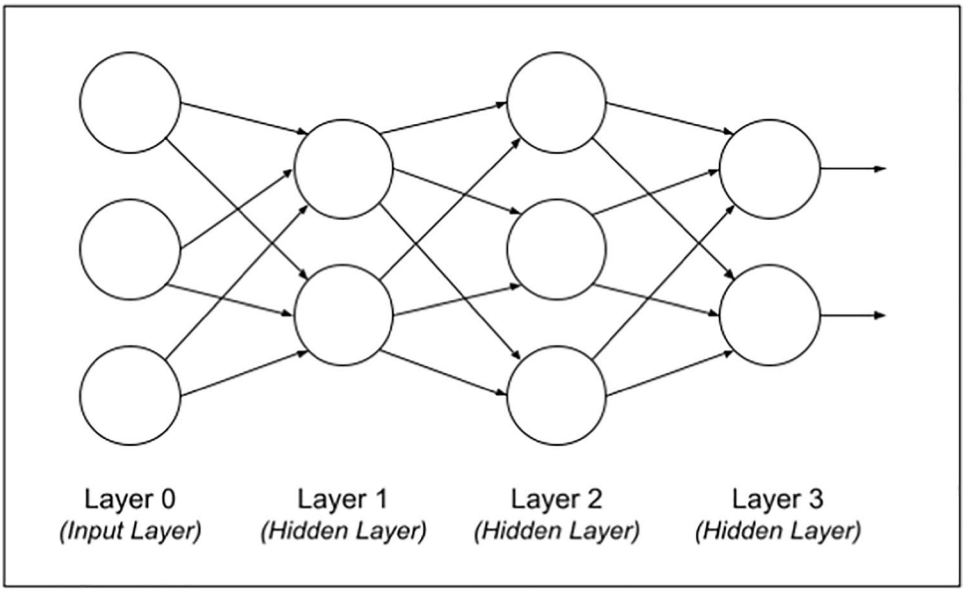


Fig. 3 A basic architecture of a feed forward neural network

composed of 3 functions f_1,f_2, and f_3 connected to form a chain, i.e., $f(x) = f_3\left(f_2\left(f_1(x)\right)\right)$. Here, x is the input to the network, f_1 is the representation of the computation of the first layer, f_2 for the second layer, and f_3 for the third. Depending on the problem at hand, more layers (i.e., more functions in the computation chain) can be added, which in turn increases the depth of the model architecture. The term deep learning reflects this concept of depth in computation.

While training, each input x is associated with a label $y = f^*(x)$. With the help of a training algorithm, the goal is to learn the weights within a network to bring $f(x)$ as close as $f^*(x)$. By doing so, the output layer gives a result that matches the desired y. For example, in a network that wants to classify the images of handwritten digits, the output layer will contain ten nodes. Each node will convey the potential output corresponding to each label (i.e., digits from 0 to 9). During training, the network will learn the weights such that when an image is shown to it, the node in the output layer corresponding to the correct label will give a high value compared to other nodes.

Loss Function

At the most basic level, a function is required to quantify how the network is performing by matching the network's output with the target goal. Such a function is called a loss function or the cost function. Ideally, the value of the loss function reduces throughout training, indicating the model is learning correctly and the given output is closer to the expected output.

Optimization

Optimization is the task of either minimizing or maximizing some function f. In the case of deep learning, we want to minimize the loss function, i.e., to reduce the gap between the actual and given output. Given below is a brief review of the calculus that is required for the optimization process.

Say we have a function $y = f(x)$, where y and x are real numbers. A derivate of this function $f'(x)$ (or $\frac{dy}{dx}$) gives the slope of the function at the point x. The slope can tell us how the value of y will change when a small change is made to the x. The derivative also gives the direction in which small changes made to x will increase the value of y. As we focus on reducing the function, the value of $f(x)$ can be lowered by moving small steps in the opposite direction of the derivative.

However, for functions with multiple inputs, partial derivatives are used. A partial derivative tells us how the function would change by a small change in an input x_i at a pointx. All of these partial derivatives are stored in a vector called the gradient, which is denoted by $\nabla_x f(x)$. The ith entry of this gradient vector tells the partial derivative of the function f w.r.t. the inputx_i.

This gradient vector gives the direction of the steepest ascent. Therefore, the value of the function f can be decreased by moving the direction of the negative gradient. This updated strategy is also known as the gradient descent method. A new point is given by Eq. (1), where x_{new} is the updated value of x that decreases the function f and β is the learning rate that controls the size of the update.

$$x_{\text{new}} = x - \beta \nabla_x f(x) \tag{2}$$

Convolutional neural Network

CNN is among the most popular network architecture used in computer vision. The convolution operation lies at the core of the CNN implementation. A predefined kernel, also known as a filter, slides across the matrix representation of input image pixels. An output is generated by multiplying and adding the kernel values with the input features. CNN architecture comprises multiple such convolutional layers, where the output from the convolution operation from the previous layer becomes an input to the next one. These kernels help to generate feature maps that contain high- and low-level features extracted from the image. These feature representations are in turn used to perform the computer vision task at hand.

The convolution operation is linear; therefore, activation functions are placed between the convolutional layers to introduce non-linearity. Some of the popular activation functions used in CNNs include sigmoid, TanH, and ReLU. Additionally, a pooling operation is applied to decrease the spatial size of representations. This operation summarizes the values within a neighborhood and replaces them with a single value. Out of the many pooling operations, the most commonly used method is max-pooling, which keeps only the maximum output from a set of neighboring values. Figure 4 illustrates a representation of a simple CNN architecture. Multiple convolutional layers with ReLU activation function and pooling operations extract features from the given food image. The feature representation thus generated is further used by a neural network to draw final predictions.

Methodology

There are several deep learning model architectures that are being used in the current times to pursue object detection. This section contains a brief discussion focusing on the frameworks that build up to the object detection algorithms used for our purpose and describes the overview of the work.

You Only Look Once

You Only Look Once (YOLO) is one of the most popular one-stage object detection frameworks (Redmon et al. 2016). Unlike the two-stage object detection methods, where the detector first proposes regions of interest and then classifies them, YOLO combines the step of region proposal and detection by acting on a dense sampling of possible locations. Owing to this, the YOLO detector can perform inference very fast compared to other two-stage methods.

Workflow of YOLO Framework

A CNN architecture is pre-trained for the specific classification task. The input image is divided into $S \times S$ grid cells. Each of the grid cells is responsible for finding the presence of the object whose center falls in that particular grid. It predicts B bounding boxes and the confidence score of each bounding box. Each grid cell also predicts K conditional class probabilities, that is the probability of a class given an object.

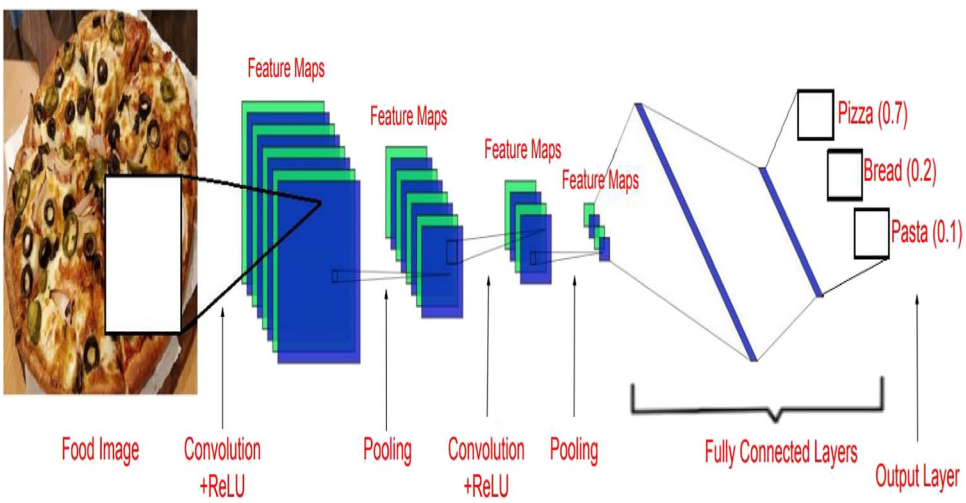


Fig. 4 Convolutional neural network (CNN) architecture

Bounding Box

Each bounding box is denoted by four values: x, y, w, h. (x, y) refer to the center coordinates of the object relative to the grid cell. (w, h) refer to the width and height, which is relative to the whole image.

Confidence Score

The confidence score indicates how confident the model is that the bounding box contains an object. It is denoted by $\Pr(\text{object}) \times \text{IoU}(\text{pred}, \text{truth})$. Here, $\Pr(\text{object})$ tells the probability of an object, and $\text{IOU}(\text{pred}, \text{truth})$ gives an intersection over the union of the ground truth with the predicted box.

Conditional Class Probabilities

The cell, when contains an object, gives the probability of the particular object belonging to all the K class labels. Irrespective of the number of the bounding boxes, the model predicts only one set containing K probabilities per cell. The dimension of the output tensor given by the model is $S \times S \times (5B + K)$.

Loss Function

YOLO promotes the model to predict accurate bounding box coordinates and match the predicted conditional class probabilities with the actual one. For the same, the loss function of YOLO can be broken down into two parts, i.e., *localization loss*, to get accurate bounding box coordinate predictions, and *classification loss* for conditional class probabilities.

$$\begin{aligned} \boldsymbol{L}_{\text{loc}} &= \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] \\ &+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \end{aligned} \tag{3}$$

$$\begin{aligned} \boldsymbol{L}_{\text{cls}} &= \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{obj}} \left(C_i - \widehat{C}_i \right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{\text{noobj}} \left(C_i - \widehat{C}_i \right)^2 \\ &+ \sum_{i=0}^{S^2} \sum_{c \in \text{classes}} \mathbb{I}_{ij}^{obj} \left(p_i(c) - \widehat{p}_i(c) \right)^2 \end{aligned} \tag{4}$$

$$L_{\text{final}} = L_{\text{loc}} + L_{\text{cls}} \tag{5}$$

In the equations, $\mathbb{I}_{ij}^{\text{obj}}$ and $\mathbb{I}_{i}^{\text{obj}}$ refer to the indicator function of whether the jth the bounding box of the grid cell i is important for object detection and whether the grid cell i contains an object, respectively. The confidence score of any grid cell i is given by C_i and the predicted confidence is given by $\widehat{C}_i$. $p_i(c)$ denotes the conditional probability of a grid cell i containing an object c from a set of given classes, while $\widehat{p}_i(c)$ tells the predicted conditional class probability.

However, there are limitations associated with YOLOv1. It is able to detect only a small number of objects; the recall is low and shows high localization error. It also fails to recognize objects that are shaped irregularly, as the bounding boxes are limited.

YOLOv2

The researchers improve upon the YOLOv1 to counter the limitations of the latter (Redmon and Farhadi 2017). To prevent the bounding box prediction from diverging too much from the center location, YOLOv2 introduced the idea of anchor boxes in the YOLO framework. The model predicts 5 coordinates per anchor box, i.e., t_x, t_y, t_h, t_w, and t_o. Assuming that the anchor box has width and height of p_w and p_h, at the grid cell which if offset from the top left corner of the image by (o_x, o_y), the corresponding predictions to the anchor box can be written as:

$$b_x = \sigma\left(t_x\right) + o_x \tag{6}$$

$$b_y = \sigma\left(t_y\right) + o_y \tag{7}$$

$$b_w = p_w e^{t_w} \tag{8}$$

$$b_h = p_h e^{t_h} \tag{9}$$

$$\Pr(\text{object}) \times \text{IOU}(b, \text{object}) = \sigma\left(t_o\right) \tag{10}$$

where $\sigma()$ is the sigmoid function, and Eq. (9) is the confidence score.

Additionally, YOLOv2 made use of a lightweighted base model of DarkNet-19, which primarily uses filters of 3×3 across 19 convolution layers and 5 max-pooling layers. It also uses global average pooling along with the use of 1×1 filers in between the 3×3 convolutions to compress the feature representation. This paper also signified the importance of using batch normalization across all the convolutional layers for better convergence.

YOLOv3

YOLOv3 (Redmon and Farhadi 2018) applied several design tricks to improve upon the YOLOv2 algorithm. It used Darknet-53 (Redmon and Farhadi 2018) with residual blocks as the backbone. YOLOv3 allowed multi-scale predictions as it added convolutional layers after the feature extractor. It made predictions at three different scales and had more bounding boxes of varying sizes. While YOLO and YOLOv2 used the sum of squared errors for classification terms, YOLOv3 used logistic regression to predict a confidence score of each bounding box.

YOLOv4

YOLOv4 (Bochkovskiy et al. 2020) improved upon the previous YOLO versions in terms of both accuracy and speed. YOLOv3 used the Darknet53 (Redmon and Farhadi 2018) architecture, which comprises 53 convolutional layers. However, in YOLOv4, the backbone used is the CSPDarknet53, which is based on the strategy of CSPNet (Wang et al. 2020).

CSPNet improves upon the DensetNet architecture (Huang et al. 2017) by partitioning the base layer's feature map into two parts. In a dense block of the YOLOv4 architecture, there are multiple convolution layers, where instead of using the output from the last layer, the output of all the previous layers is collected. Therefore, at every layer, the number of feature maps is increased based on the number of the previous layer, which refers to the growth rate. Figure 5a shows the structure of a dense block with four dense layers within it. Each layer concatenates the output from the previous layers. The transition block consists of convolution and pooling. On the contrary, CSPNet splits the input into two parts, where one part goes into the dense block and the other part is aggregated with the results of the first part, as shown in Fig. 5b. Figure 6 contains the illustration to compare the dense block from DenseNet and CSPNet.

Instead of directly passing the features from the backbone into the head, a neck block with additional layers is added in between. The neck block enriches the information fed to the head block by using the neighboring feature maps from the bottom-up and top-down streams. By doing so, the input to the head has both spatial and semantic information, allowing it to make better predictions. Finally, the head block makes the dense prediction, which is used as the network's output. The head block is the same as used in YOLOv3. For every grid cell that the image is divided into, the output from the head block consists of bounding box coordinates and probability classes.

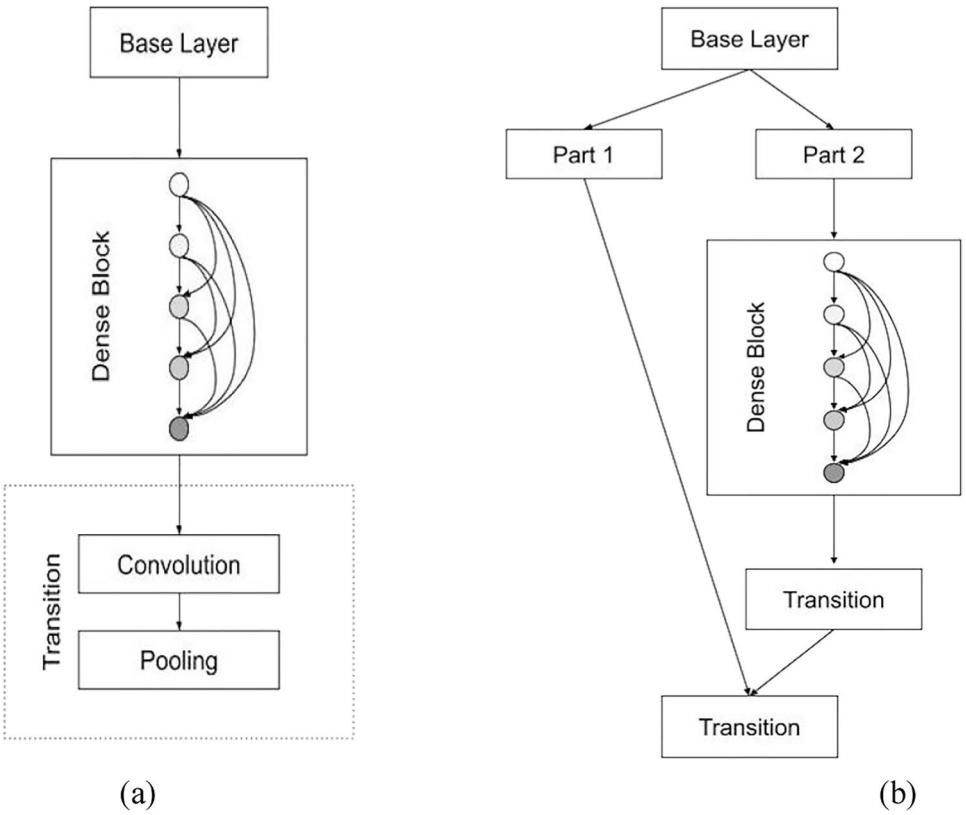
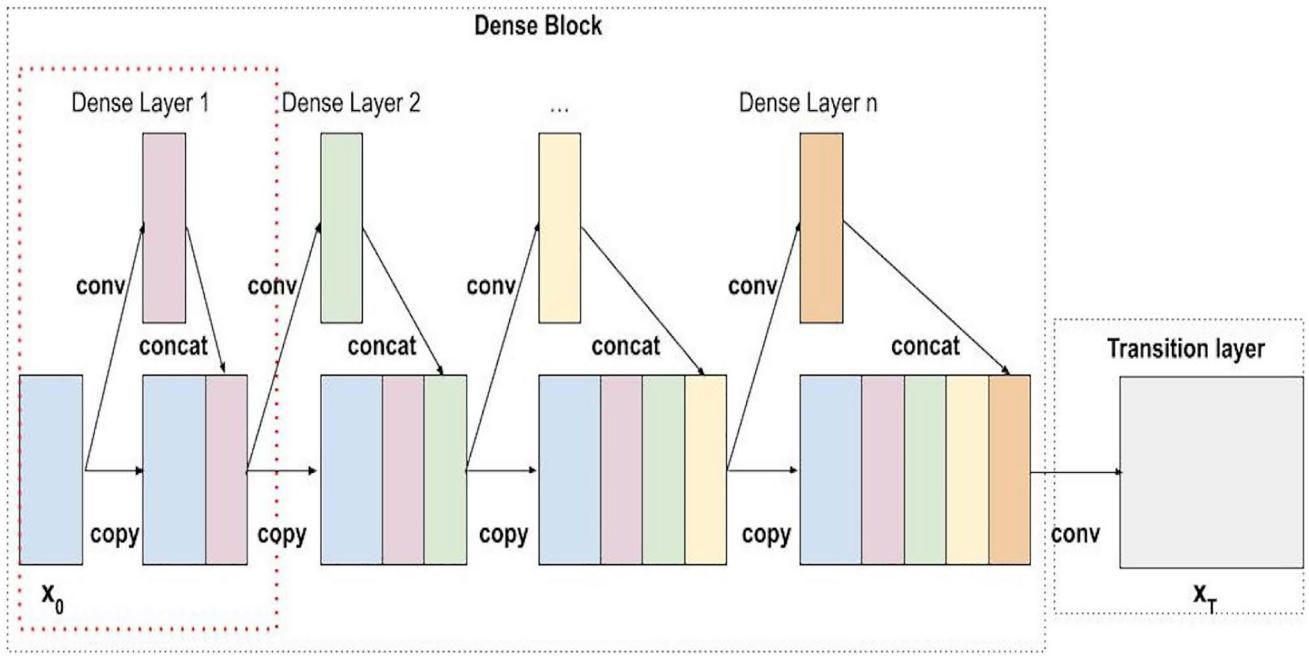
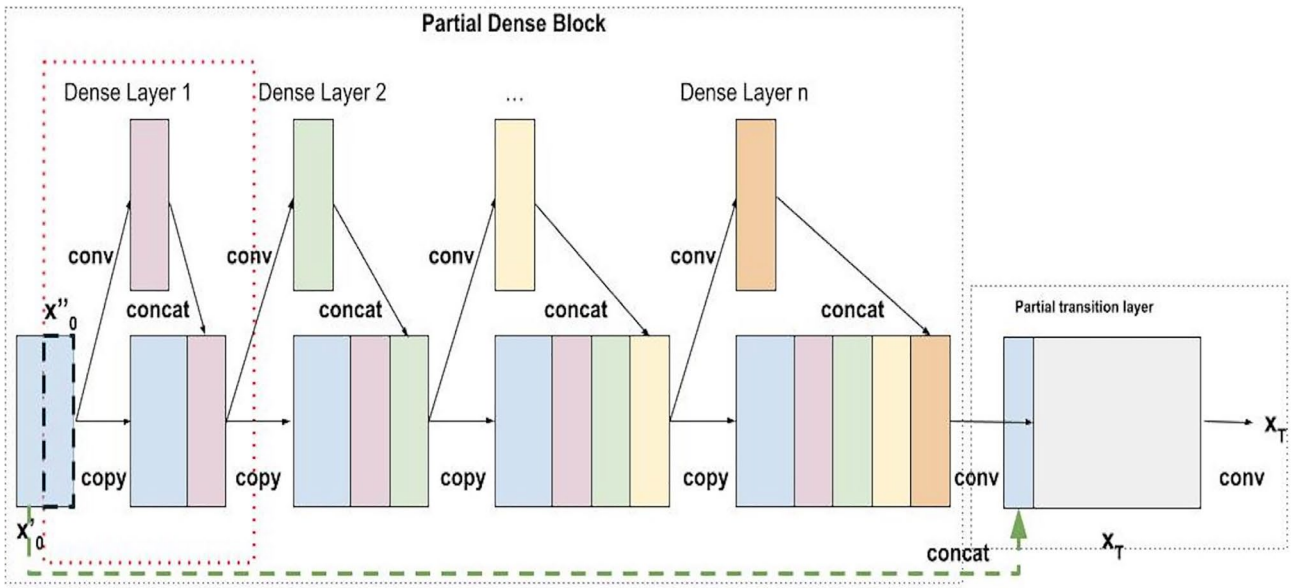


Fig. 5 Representation of a dense block in **a** DenseNet and **b** CSPNet



(a)



(b)

Fig. 6 Network illustration comparing **a** DenseNet with **b** CSPNet

YOLOv4 introduced the bag-of-freebie and the bag-of-specials with a focus to improve the performance of the algorithm. The bag-of-freebie contains the improvements that are made during the training process to increase the accuracy, without harming the inference speed, whereas bag-of-specials focuses on improving the network architecture to get a boost in performance.

YOLOv5

YOLOv5 is a lightweight version of the YOLO family, released nearly a month after YOLOv4. This version was publicly released on Github (https://github.com/ultralytics/yolov5) and is developed and maintained by Ultralytics. Because YOLOv5 is written in the popular Pytorch framework, it continues to grow from the contributions made by the community. Even though there is not much difference in the theory of YOLOv4 and YOLOv5, YOLOv5 is a popular choice for object detection due to the remarkable engineering efforts made by the developers.

Like YOLOv4, the architecture of YOLOv5 can also be divided into three parts, namely the backbone, neck, and head. CSPDarknet is used as the backbone of the architecture, which counters the problem of repeated gradient information in large-scale backbones. It integrates the gradient changes into the feature maps, which reduces the model size and improves the accuracy and inference speed of the model. The focus layer replaces the first three layers of the YOLOv3's architecture with a single layer in YOLOv5, with the primary goal of reducing the number of layers and increasing the forward and backward speed, without compromising the performance of the network.

After the feature extraction from the backbone, YOLOv5 deploys a modified path aggregation network (PANet) (Liu et al. 2018a, b) as the neck architecture. The core principle is to aggregate the information from the backbone to get improved results. The initial layers of a deep neural network extract the low-level features from an input, which are lost when the information flows ahead. However, such features can play an important role in object detection. PANet uses a modified feature pyramid network structure with a bottom-up path, to allow the low-level, fine-grained information to reach the top layers efficiently. Additionally, a spatial pyramid pooling layer (He et al. 2014) is added to preserve the output spatial dimension while feature maps of different sizes are concatenated. Finally, the head of YOLOv5 architecture makes multi-scale predictions on different sizes of feature maps to allow the model to detect objects of varying sizes within an image. An overview of the YOLOv5 architecture is given in Fig. 7.

YOLO-R

YOLOR (Wang et al. 2021) or You only learn one representation is a recent work that aims to perform multiple tasks including object detection using a single unified model. The core idea of the model is to collect both implicit and explicit information simultaneously from one representation. Explicit information refers to coarse details of an image that can be known from the shallow layers of a network. On the other hand, implicit information refers to the finer details made available in later layers of the network. Given an input, YOLOR extracts both explicit and implicit information and then combines the two representations for further tasks.

A general training function of a neural network can be summarized as follows:

$$y = f_\theta(x) + \text{error} \tag{11}$$

where, for a given input x, we seek to find the parameters θ of the function f, which reduces the error term. However, the solution space thus received is helpful for a specific task t_i. The solution will probably be invariant to other potential tasks from a set $T = \{t_1, t_2, \ldots, t_t\}$. In order to build a multi-purpose representation that can be used for all the tasks in T, the error term is modeled further to give a modified training function as:

$$y = f_\theta(x) + \text{error} + h_\delta\left(g_{\text{exp}}(x), g_{\text{imp}}(z)\right) \tag{12}$$

where the expressions g_{exp} and g_{imp} are the operations to model the explicit and implicit error, respectively, from the input x and latent representation z. To further combine the relevant information from the implicit and explicit knowledge learning, h_δ operation is used.

The explicit information is gained from the input x, while the implicit information is learnt using the latent representation z. So, by using certain existing methods, Eq. (11) can be integrated as:

$$y = f_\theta(x) h_\delta(z) \tag{13}$$

where refers to the possible operations like addition, multiplication, and concatenation. Finally, to perform multiple tasks, the derivation process of the error term can be stated as the following equation:

$$F(x, \theta, Z, \Delta, Y, \Phi) = 0 \tag{14}$$

where $Z = \{z_1, z_2, \ldots, z_t\}$ refers to the set containing the implicit latent representation for the t different tasks in T, Δ refers to the parameters that build the implicit representation from Z, and Φ refers to the parameters that generate the final output from the various combination of implicit and explicit information.

Figure 8 holds a graphical representation of the core architecture of YOLOR. The analyzer learns a representation of the input image. The explicit knowledge is gained using the input image, whereas implicit knowledge is learnt using latent representations. The implicit knowledge is of key importance for learning generalisable representations. They can be modeled as a vector (a direct representation),

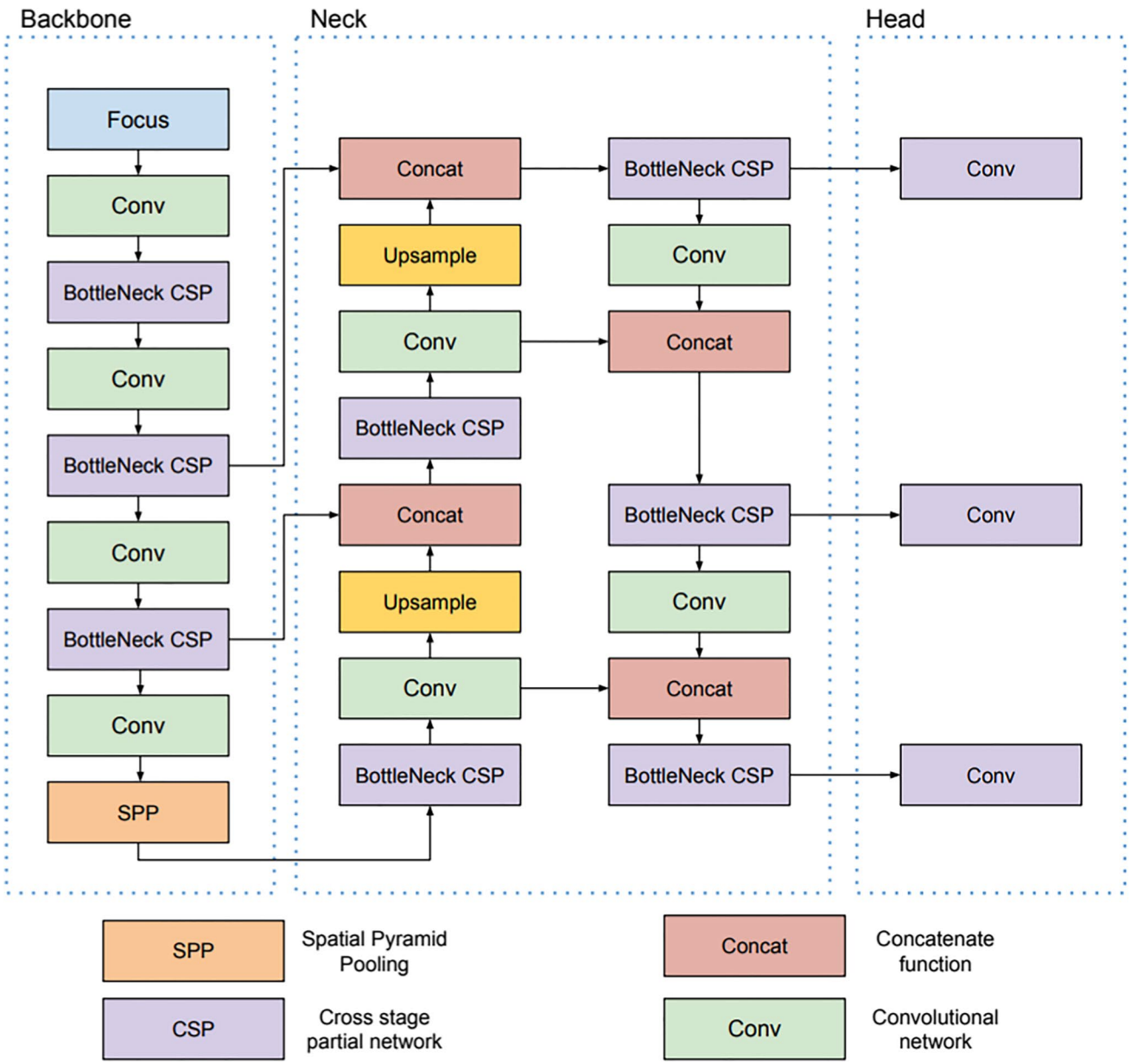


Fig. 7 Network architecture of YOLOv5

neural network (learning linear combination on the previous vector), or matrix factorization (multiple vectors combined to represent implicit knowledge). At last, the representations are combined and a final discriminator gives the corresponding output depending upon the task at hand.

System Overview

The basic flow of our approach is shown in Fig. 9. Our work can be divided into two stages, dataset preparation, and object detection model training. In the dataset preparation stage, we collect the relevant food images from the web and annotate bounding boxes around the food items to be further used for object detection training. More details regarding the dataset and its preparation can be found in the "Methodology" section. During the second stage, we define a model architecture along with relevant loss function, optimizer, and hyperparameters. This model makes predictions on the training set and updates itself to minimize the loss value. After the allotted training steps, the trained model draws inference on the test set, which conveys the efficacy of the trained model on unseen data.

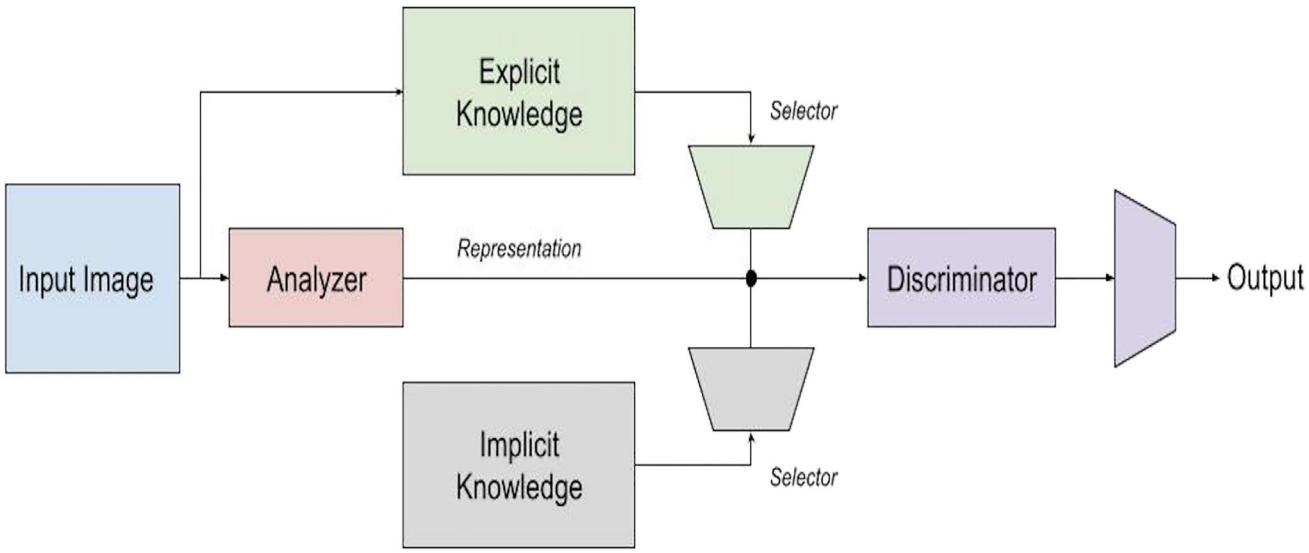


Fig. 8 The unified network of YOLOR for learning representations for multiple tasks

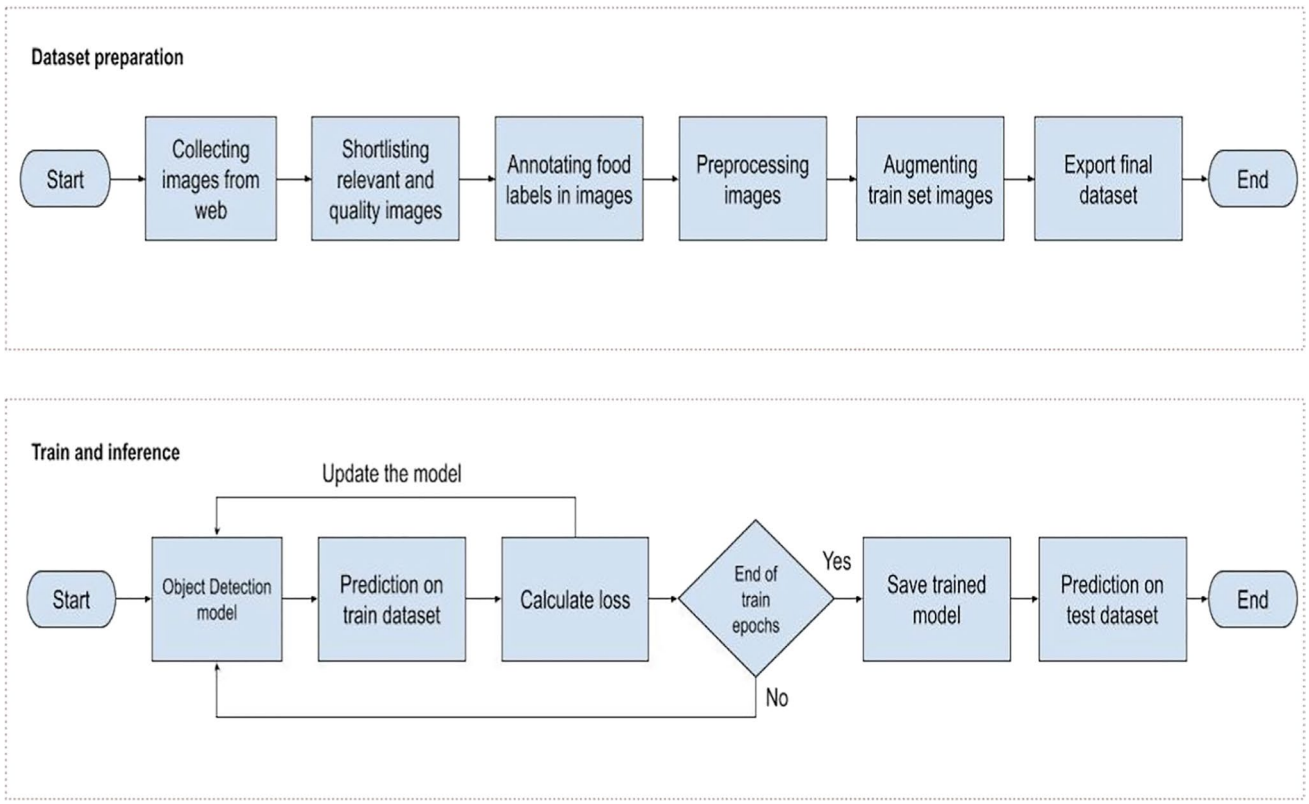


Fig. 9 The flow of approach

Dataset Description

We have modeled our dataset named Allergen30 (https://universe.roboflow.com/allergen30) for the work. The dataset consists of images of food items that can cause an allergic reaction, along with annotation box(es) per image that specifically locates the position of the food item. We shortlist the food items primarily based on three criteria: (a) they contain high traces of the commonly occurring allergen mentioned in the "Food intolerances" section, (b)

Table 6 Description of each class label present in the dataset Allergen30 (https://universe.roboflow.com/allergen30)

S. no	Allergen	Food label	Description
1	Ovomucoid	Egg	Images of egg with yolk (e.g., sunny side up eggs)
2	Ovomucoid	Whole egg boiled	Images of soft and hard boiled eggs
3	Lactose | histamine	Milk	Images of milk in a glass
4	Lactose	Ice-cream	Images of ice cream scoops
5	Lactose	Cheese	Images of Swiss cheese
6	Lactose | caffeine	Milk based beverage	Images of tea/coffee with milk in a cup
7	Lactose | caffeine	Chocolate	Images of chocolate bars
8	Caffeine	Non milk based beverage	Images of tea/milk without milk in a cup
9	Histamine | gluten	Cooked Meat	Images of cooked meat
10	Histamine | gluten	Raw Meat	Images of raw meat
11	Histamine	Alcohol	Images of alcohol bottles
12	Histamine	Alcohol glass	Images of wine glasses with alcohol
13	Histamine	Spinach	Images of spinach bundle
14	Histamine	Avocado	Images of sliced avocado
15	Histamine	Eggplant	Images of eggplant
16	Salicylate	Blueberry	Images of blueberry
17	Salicylate	Blackberry	Images of blackberry
18	Salicylate	Strawberry	Images of strawberry
19	Salicylate	Pineapple	Images of pineapple
20	Salicylate	Capsicum	Images of bell pepper
21	Salicylate	Mushroom	Images of mushrooms
22	Salicylate	Dates	Images of dates
23	Salicylate	Almonds	Images of almonds
24	Salicylate	Pistachios	Images of pistachios
25	Salicylate	Tomato	Images of tomato and tomato slices
26	Gluten	Roti	Images of roti
27	Gluten	Pasta	Images of one serving of penne pasta
28	Gluten	Bread	Images of bread slices
29	Gluten	Bread loaf	Images of bread loaf
30	Gluten	Pizza	Images of pizza and pizza slices

they are commonly consumed on a daily basis, and most importantly, (c) they can be visually distinguishable, as we primarily rely on computer vision methods to detect the presence of food items in an image.

Collection

We used search engines (Google and Bing) to crawl and look for suitable images using JavaScript queries for each food item from the list created. The images with incomplete RGB channels were removed, and the images collected from different search engines were compiled. After merging, several duplicate images were encountered. We implemented image hashing to conduct the detection of such duplicate images and removed them from the dataset. When downloading images from search engines, many images were irrelevant to the purpose, especially the ones with a lot of text in them. We deployed the EAST text detector to segregate such images (Zhou et al. 2017). Finally, a comprehensive manual inspection was conducted to ensure the relevancy of images in the dataset.

Annotation

After collecting the dataset, the images were annotated on Roboflow's (https://roboflow.com/) online platform using their self-serve annotation tool. The platform allows the annotation of the entire dataset with ease without downloading an external program. The labeling interface required creating a bounding box around the concerned food item in a given image, which was achieved by clicking and dragging the box around the dimension of the food item. The class selector tool was then used to choose the corresponding label of the food item in the bounding box. To speed up the annotations process, we also made use of Roboflow's model assist labeling tool. This tool

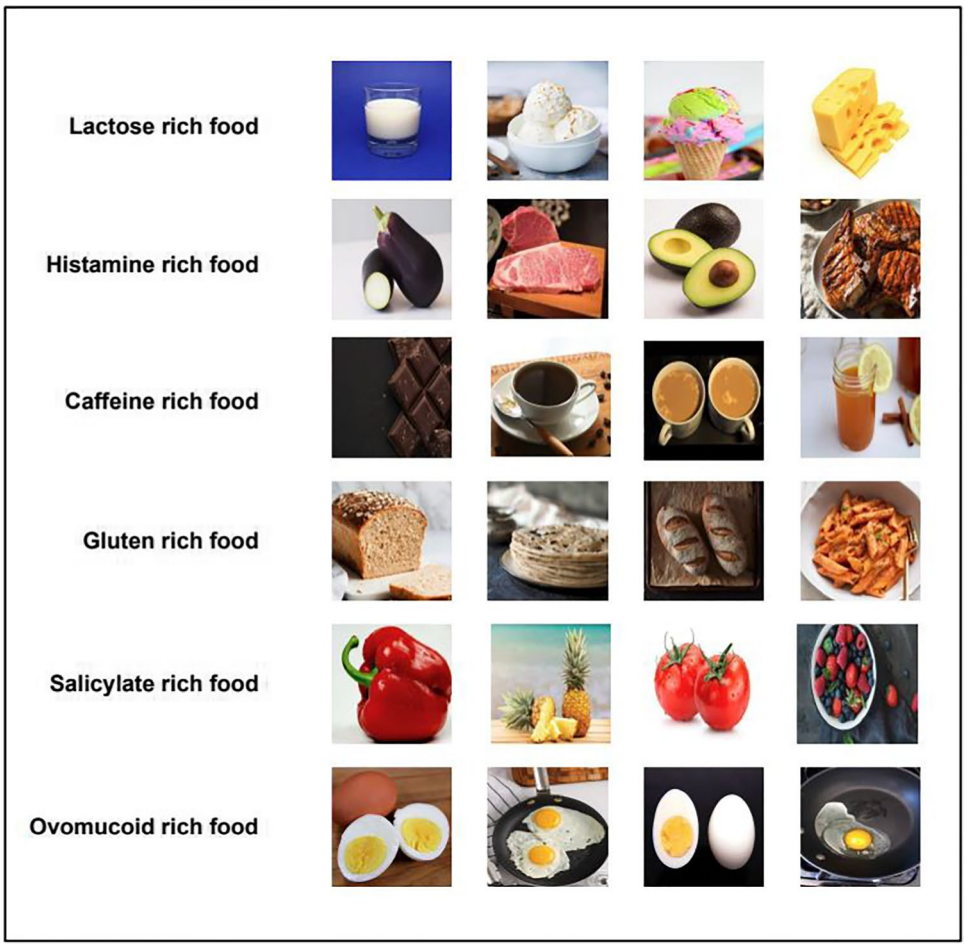


Fig. 10 Common food intolerances

allowed a model to train on a sample of the annotated data. This trained model automatically added annotation when more images were added to the dataset. The annotations thus made were then thoroughly examined manually for accuracy and quality. The following link can be used to download the dataset and to gather more information regarding the number of annotations per label (https://universe.roboflow.com/allergen30).

Preprocessing and Augmentation

Following the annotation of all the images, the dataset is divided into a train (70%), validation (20%), and test (10%) set. The images are resized into 416 × 416 and are auto-oriented. Auto-orientation manages the EXIF data to allow the images to be displayed the same way they are stored. To further assist during the training part, every image in the train set is augmented three times based on specific settings. Augmenting helps the model by providing more diverse sample images, making it robust and generalizable. We perform offline augmentation along with generating augmented samples during training to increase the model's reproducibility and decrease train time and costs. We apply random shear to distort an image across the vertical and horizontal axis and randomly rotate an image 90 degrees clockwise, counter-clockwise, and upside down.

Impact of the Dataset

The Allergen30 dataset is among the first research attempts to make use of the current deep learning–based computer vision methods to identify the presence of allergen-based food using images/videos as an input. These methods can help the larger masses prevent reactions that can be avoided by simply being aware of the allergens. Because the focus is specific to a problem, it eliminates the need to train a model on extremely large amounts of food images. At the same time, it also encourages the research of computer vision methods to better learn the otherwise difficult visual cues related to food items.

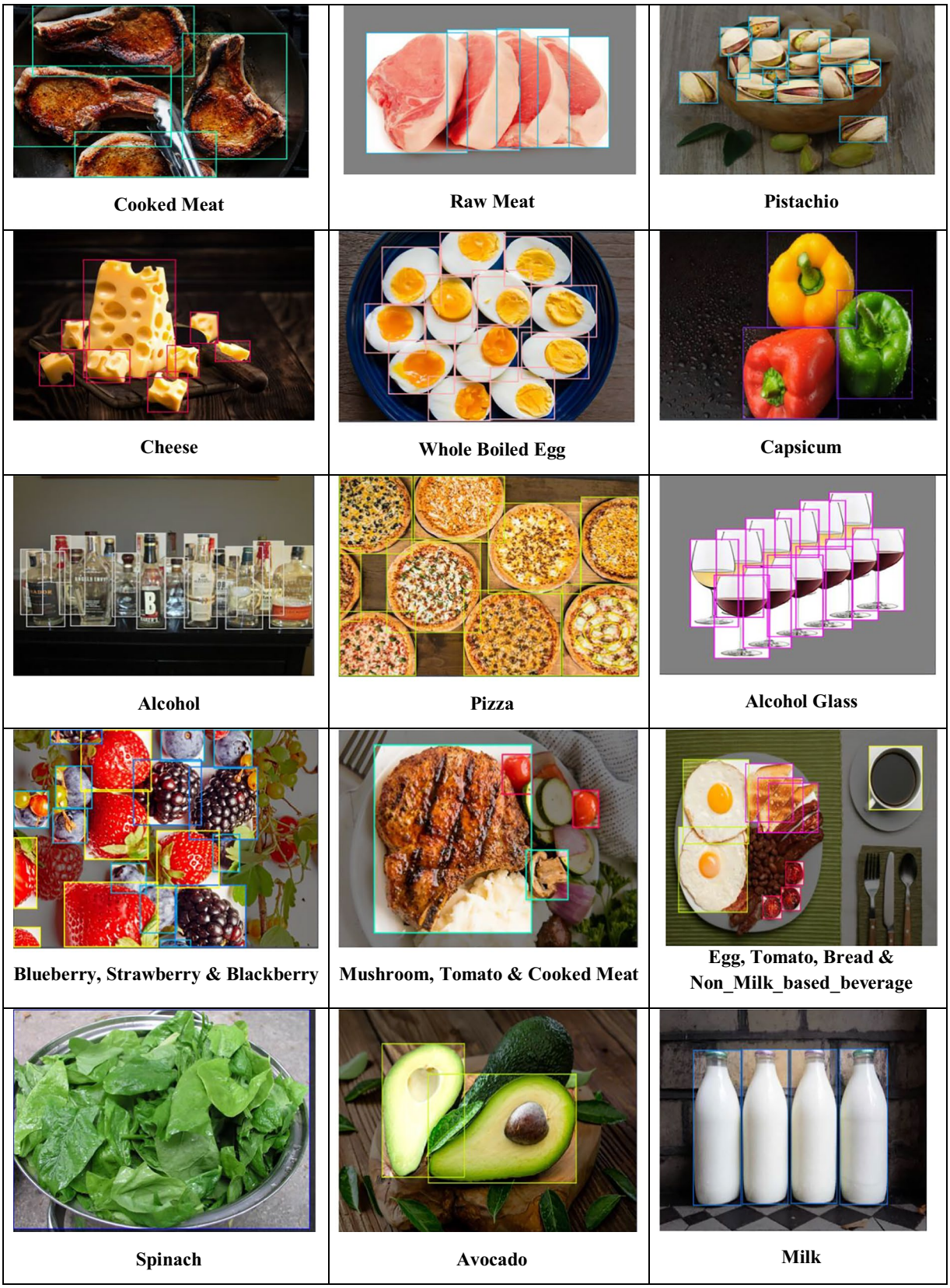
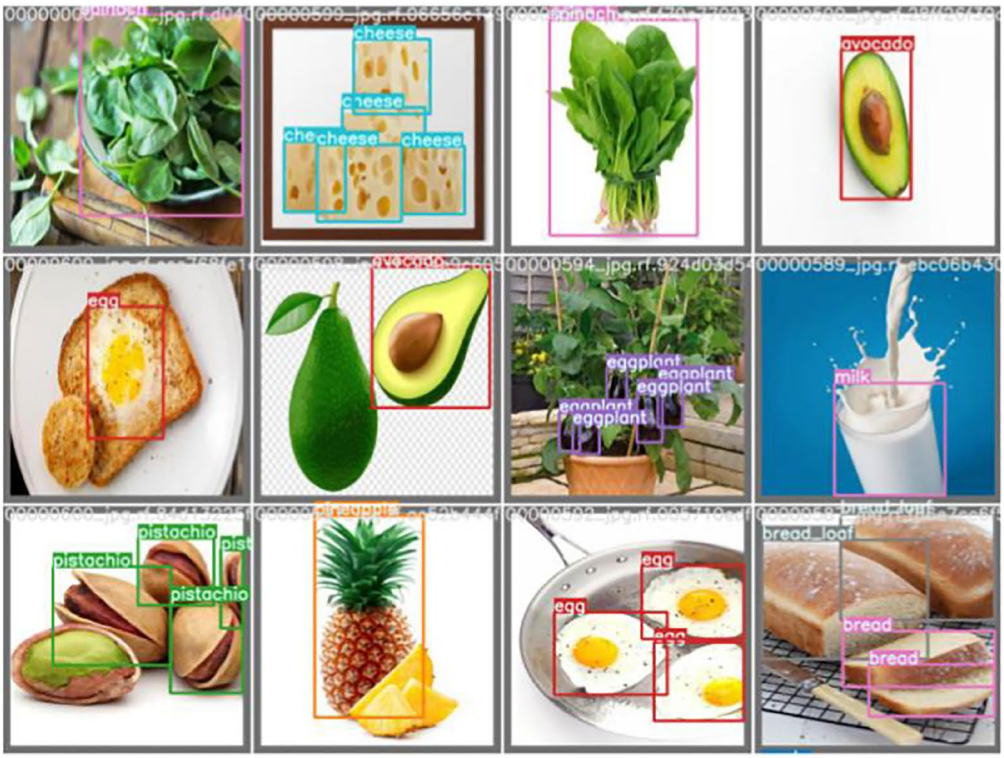
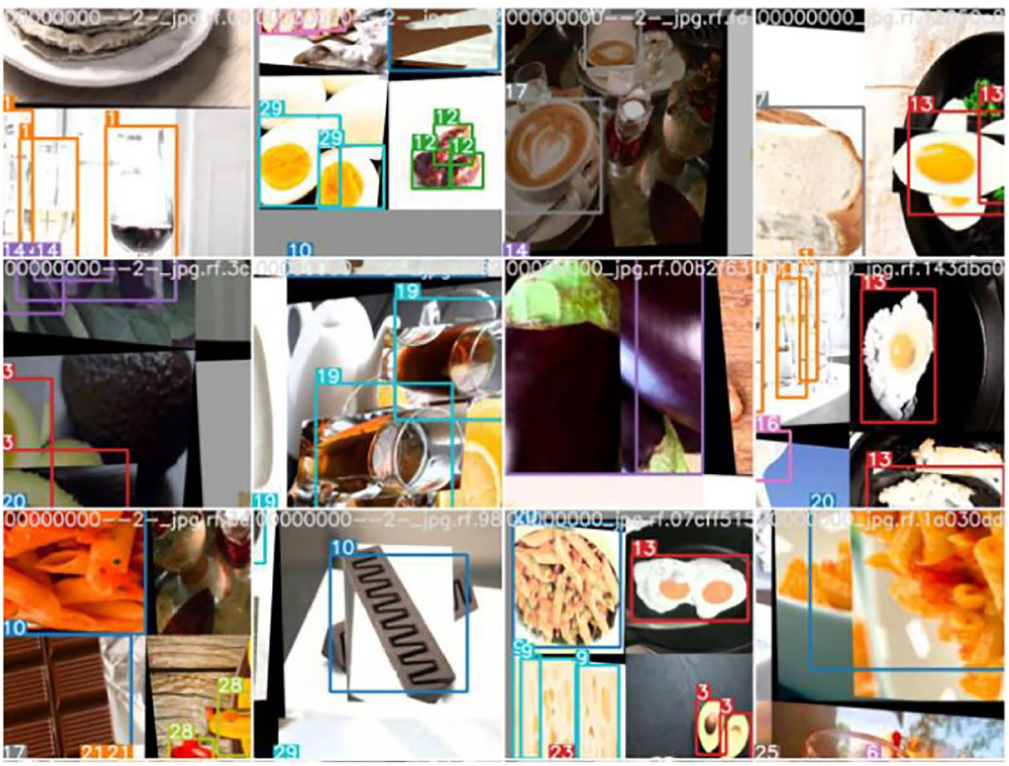


Fig. 11 Samples of annotated images from Allergen30 Dataset



(a)



(b)

Fig. 12 Comparing the training images before and after applying augmentations. **a** Sample of training images before augmentation. **b** Sample of training images after augmentation

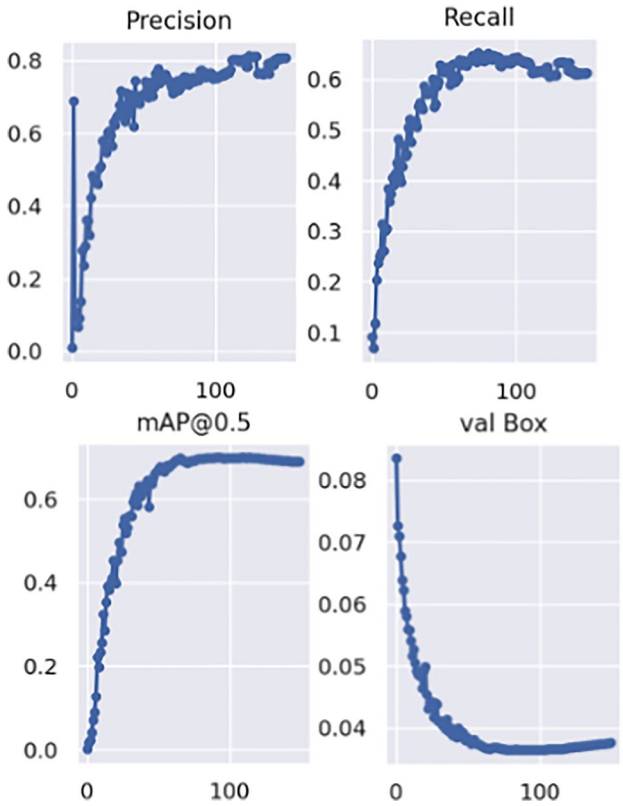


Fig. 13 Training summary of YOLOv5m across 150 epochs

Data Statistics

The Allergen30 dataset contains more than 6000 images with 18,000 + annotations spreading across 30 food labels. Table 6 shows the description of each class label present in our dataset. Common food intolerances are added in Fig. 10. Sample annotations for sample food items are added in Fig. 11.

Results and Discussion

Training Details

We deploy YOLOv5 and YOLOR training algorithms to train a model for detecting the presence of allergen-based food items by learning from our custom-built dataset. The developers of the YOLOv5 published five variants, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These variants vary in their size, with YOLOv5x being the largest and YOLOv5s the smallest. Each variant produces different detection accuracy and performance. Within YOLOv5, we compare the results among the YOLOv5s, YOLOv5m, and YOLOv5l variants, and for YOLOR, we train the YOLOR-P6 variant. We use the GoogleColab Notebook available at Roboflow's model zoo (https://models.roboflow.com/) for training and testing purposes. These notebooks are based on the original repositories associated with YOLOv5 and YOLOR.

Table 7 Evaluation of performance of various models on test set

	Precision	Recall	F1 score	mAP	Inference speed
YOLOv5s	0.801	0.681	0.7361	0.747	5.0 ms
YOLOv5m	0.861	0.679	0.7592	0.749	7.4 ms
YOLOv5l	0.845	0.707	0.7698	0.766	8.5 ms
YOLOR	0.830	0.741	0.7829	0.811	8.4 ms

During the course of the training, the models train on the train set and evaluate themselves on the validation set. The best model received during the training phase is stored, which is used to draw inference on the test set. For training the models, we set the input image size as 416×416 and train them for 150 epochs with a batch size of 64. We initialize the models with COCO pre-trained weights to avoid the model learning from scratch. Additionally, mosaic augmentation (Hao and Zhili 2020) is applied while training. Mosaic augmentation combines multiple training images as one based on certain ratios. Sample training images before and after applying the augmentation are compared in Fig. 12.

Figure 13 shows the change in precision, recall, and mean average precision (mAP) (y-axis) over 150 training epochs (x-axis) for the YOLOv5m model. As the training progresses, the bounding box regression loss decreases on the validation set (Val box). On the other hand, the precision, recall, and mAP increased during the course of training, indicating that the model learns to make better predictions over the training epochs. This trend in training summary is witnessed for all the models.

Evaluation Metrics

The efficacy of object detection models is studied using various evaluation metrics, namely precision, recall, F1 score, and mean average precision. mAP is the most popular among other evaluation metrics. Intersection over union (IoU) indicates the overlap of the predicted bounding box with the given ground truth. Precision measures the fraction of correct positive predictions with respect to all observations. Recall measures the fraction of correct positive predictions with respect to all ground truth. In object detection, precision and recall are calculated using the IoU threshold. F1 score is the harmonic mean or a weighted average of recall and precision. The average precision of a particular class is the average precision of all possible decision thresholds in that class. mAP is the mean of average precision of all classes. We also report the inference speed (in ms) as an evaluation metric. It refers to the time taken by the model to make a prediction for a given input image. This inference speed accounts for the

Fig. 14 Dates missed by YOLOv5s (**a**) is picked up by YOLOR-P6 (**b**)

Fig. 15 YOLOv5m (**a**) overlooks majority of almonds while YOLOv5l (**b**) detects them all

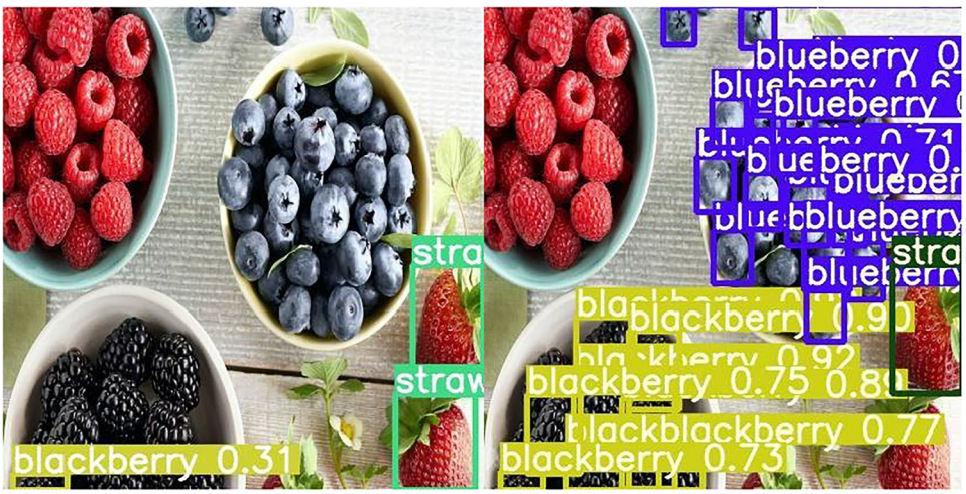


Fig. 16 YOLOv5m (**a**) fails to detect different berries while YOLOR-P6 (**b**) locates large number of berries

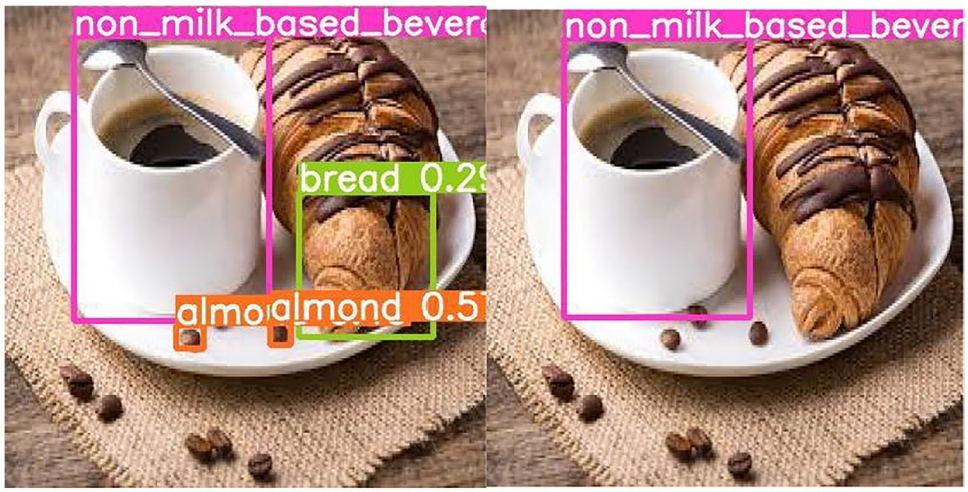


Fig. 17 YOLOv5s (**a**) gives more false positives as compared to YOLOv5l (**b**)

time taken for non-maxima suppression (NMS) as well. The above evaluation metrics equations are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{F1score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

$$\text{Mean average precision (mAP)} = \frac{1}{n} \times \sum_{k=1}^{n} \text{AP}_{\text{k}} \tag{18}$$

where TP is representing the true positive; FP is representing the false positive; FN is representing the false negative; AP_k is representing the average precision of class k; and n is representing the number of classes.

Inference

The best model from the training is used to draw inference on the test set. We evaluate the performance using precision, recall, mAP@0.5 (mAP), and inference speed (in ms). The IOU threshold for NMS is set to 0.6 and the confidence threshold is set to 0.001. The inference is drawn on the Tesla P-100-PCIE-16 GB device with an image size of 416×416 and a batch size of 64. Table 7 shows a detailed analysis of different models trained.

We observe that the smaller models (YOLOv5s and YOLOv5m) make less accurate predictions when compared to the other larger models (YOLOv5l and YOLOR-P6). YOLOv5s and YOLOv5m have an mAP score of 0.747

Fig. 18 YOLOv5s (**a**) confuses roti with pizza which is rectified by YOLOv5l (**b**)

Table 8 Evaluation of performance of the YOLOv5l model on across all the labels

Food class	Precision	Recall	F1-Score	mAP
Alcohol	0.839	0.795	0.816	0.851
Alcohol glass	0.907	0.974	0.939	0.971
Almond	0.75	0.57	0.647	0.677
Avocado	0.945	0.988	0.966	0.993
Blackberry	0.843	0.61	0.707	0.752
Blueberry	1	0.059	0.111	0.213
Bread	0.676	0.664	0.669	0.673
Bread loaf	0.72	0.806	0.760	0.812
Capsicum	0.907	0.769	0.832	0.865
Cheese	0.837	0.8	0.818	0.81
Chocolate	0.858	0.593	0.701	0.667
Cooked meat	0.8	0.778	0.788	0.829
Dates	0.944	0.705	0.807	0.839
Egg	0.836	0.96	0.893	0.959
Eggplant	0.879	0.731	0.798	0.826
Ice-cream	0.82	0.712	0.762	0.751
Milk	0.731	0.654	0.690	0.709
Milk-based beverage	0.863	0.97	0.913	0.963
Mushroom	0.835	0.0986	0.176	0.247
Nonmilk-based beverage	0.828	0.816	0.822	0.865
Pasta	0.947	1	0.973	0.994
Pineapple	0.83	0.708	0.764	0.769
Pistachios	0.882	0.672	0.763	0.76
Pizza	0.931	0.736	0.822	0.861
Raw meat	0.822	0.895	0.857	0.905
Roti	0.828	0.515	0.635	0.611
Spinach	0.988	0.909	0.947	0.988
Strawberry	0.47	0.289	0.358	0.251
Tomato	0.866	0.68	0.762	0.774
Whole egg boiled	0.962	0.759	0.848	0.81

and 0.749, respectively, whereas YOLOv5l and YOLOR-P6 model give an mAP of 0.766 and 0.811, respectively. However, the inference speed is faster for smaller models than the larger models. YOLOv5s have an inference time of 5.0 ms as compared to the 8.4-ms inference time of the YOLOR-P6 model.

A higher recall value is observed in the larger models; i.e., they are more able in detecting the presence of all the food items in an image as compared to the smaller models. Especially for food items like almonds, berries, and dates, which are present in large quantities within a single image, YOLOv5l and YOLOR-P6 perform better at detecting these labels than the smaller variants. In Fig. 14, YOLOR-P6 succeeds in picking up the presence of the dates that were left by YOLOv5s. Similarly, Fig. 15 depicts that YOLOv5l is able to find almost all the almonds within the image, performing better than the smaller variant of YOLOv5m. A similar trend is seen in Fig. 16, where YOLOR-P6 detects the vast numbers of blueberries and blackberries, which are otherwise missed by YOLOv5m.

Many samples verify that YOLOv5s give more false positives than YOLOv5l and YOLOR-P6. The models are primarily inclined to give false positives for food labels with similar visual cues. For example, in Fig. 17, YOLOv5s confuses the coffee beans for almonds and a small part of the croissant for bread, whereas the YOLOv5l model rectifies this mistake. It is noteworthy that the model is not trained to detect croissants as bread. Similarly, in Fig. 18, the YOLOv5l model correctly identifies the food label as roti, whereas the YOLOv5s variant confuses it with the similar-looking pizza.

Table 8 shows the performance of the YOLOv5l model across all the labels. Class labels that record-high mAP primarily include food labels with more training samples and more distinct visual characteristics than others along with a

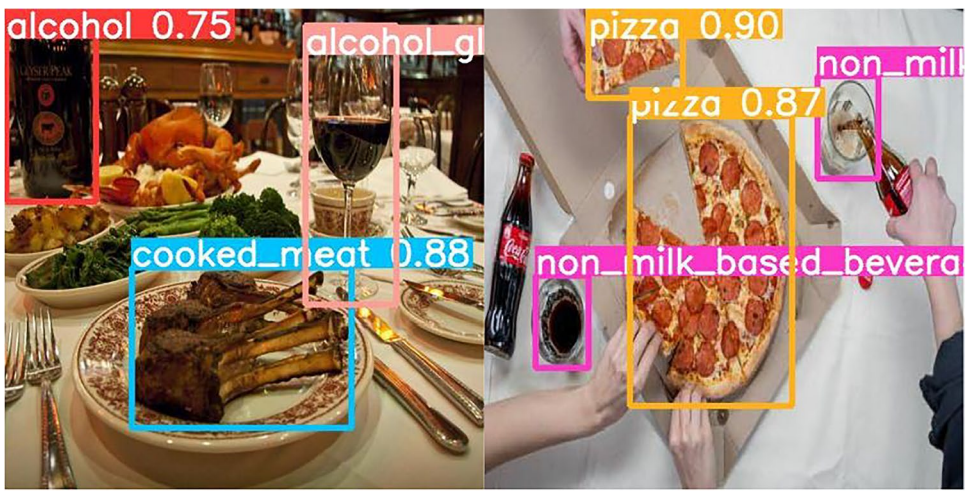


Fig. 19 Model trained on our dataset is able to detect multiple labels from a variety of food items. **a** Cooked meat, alcohol; **b** nonmilk-based beverage, pizza

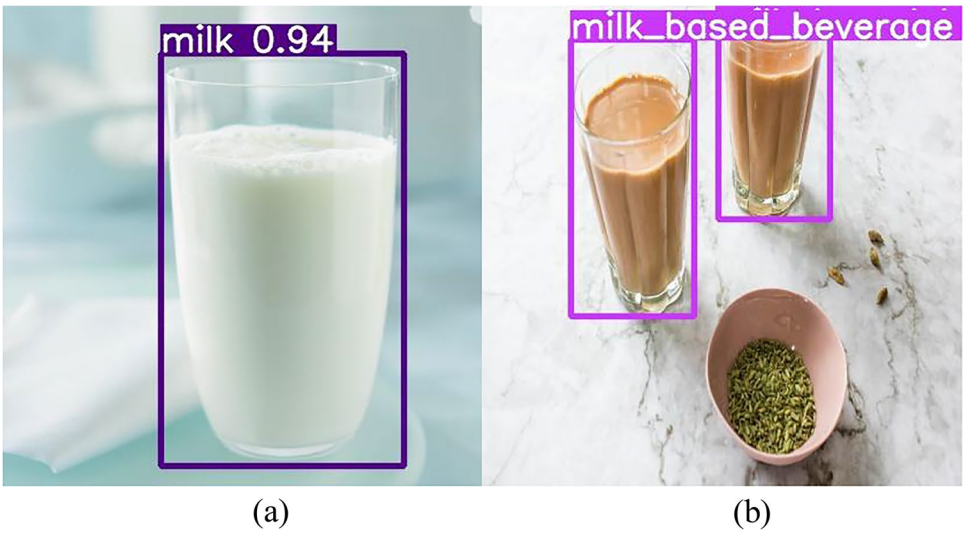


Fig. 20 Even though they are served in similar glass, the model is able to discriminate between **a** milk and **b** milk-based beverage

consistent representation within the label. For example, class labels like egg, spinach, and raw meat enjoy an mAP of more than 0.9, whereas labels like mushrooms suffer in terms of mAP because of their variety found in training images. Food labels like strawberry, blueberry, blackberry, and almonds, which are found in large numbers within one image, fail to gain a high recall. On the other hand, food labels like pasta, and avocado, which can be easily and distinctly identified within an image, report a higher recall. Overall, the model performs decently well with an overall precision and recall of 0.845 and 0.707, respectively, with an F1 score of 0.769 and mAP of 0.766.

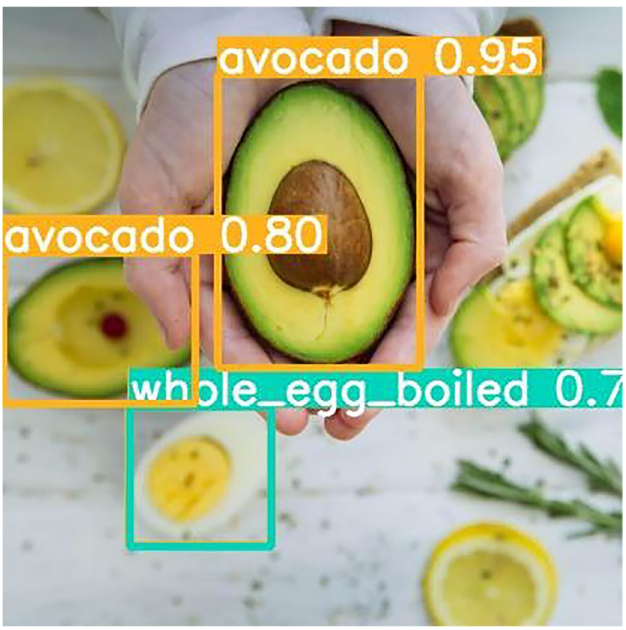


Fig. 21 Model is able to differentiate between the similar looking avocado and boiled egg

Successful Predictions with the Allergen30 dataset

In this subsection, we further analyze the predictions made by YOLOv5l trained on our Allergen30 dataset. The model is able to detect multiple labels from images with a variety of food items. In a usual meal setting, it is common to pair meat with wine, which is rich in histamine. In Fig. 19a, the presence of an alcohol bottle and wine glass is successfully detected, along with the meat served on the plate. At the same time, the model successfully ignores the empty glasses of wine from the background. Similarly, in Fig. 19b, the model picks up the pizza and the slice, as well as the glasses with a soft drink. Pizza is commonly served with cheese and meat, making it rich in gluten, lactose, and histamine, whereas soft drinks contain high amounts of caffeine.

Additionally, our model does decently well in discriminating between visually similar food items. For example, in Fig. 20a, the model is able to detect a glass of milk, whereas, in Fig. 20b, the model correctly labels the two glasses as a milk-based beverage (tea and coffee made with milk), even though they are all served in a similar-looking glass. Milk-based items are rich in lactose, and beverages like tea and coffee, when made with milk, contain both lactose and caffeine. Similarly, despite the similar visual cues, the model is able to pick the presence of both avocado and boiled egg in Fig. 21. Eggs can cause a reaction due to the presence of ovomucoid, and avocado can trigger a reaction because of the high amount of salicylate. Moreover, the trained model is capable of detecting food items of varying sizes from an image as well. Figure 22a and b show that the model detects the presence of different sizes of histamine-rich eggplants and salicylate-rich pineapple, respectively.

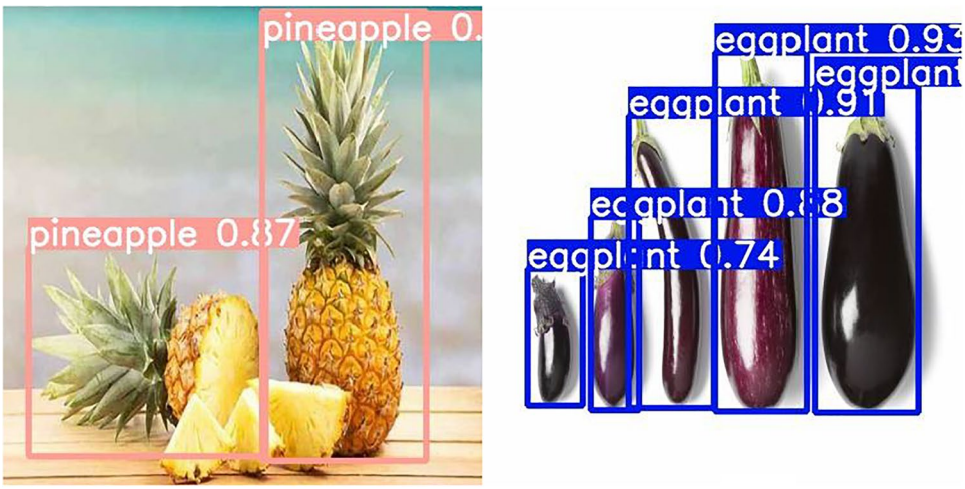


Fig. 22 Model successfully detects **a** pineapple and **b** eggplant of varying sizes

Our quantitative and qualitative analysis shows that current state-of-the-art object detection models trained on our Allergen30 dataset perform reasonably well in achieving our goal of detecting the presence of commonly consumed allergen-based food items.

Choice of Model

The choice of model to be used for deployment becomes critical for object detection. We see that the models that perform better in terms of predictions are often slow for inference and vice versa. YOLOv5 is smaller than the previous YOLO family architectures. Compared with the immediate predecessor, YOLOv5 is around 90% smaller than YOLOv4. Even though the model is extremely fast and lightweight, the accuracy is on par with the YOLOv4 benchmark. Apart from YOLOv5, the recent YOLOR algorithm too achieves a high accuracy score on the COCO dataset. This makes YOLOv5 and YOLOR promising candidates for deploying an object detection model.

Within the variant of such object detection algorithms, one has to evaluate the environment of deployment to make an informed decision about the model to be used. In situations where object detection models are used to detect objects in real time, i.e., counting the number of cars from a live traffic feed or detecting people for security purposes during live video surveillance, it is more apt to pick a small size model. On the other hand, applications that require accurate predictions, even at a cost of time, prefer larger variants. Such applications include detecting tumors from MRI scans.

The application of detection of allergen-based food can be deployed in various scenarios. It can be used in a mobile application that requests an image from the user and predicts the presence of any allergen-based food item within the image. Such a deployment opens up the doors for models like YOLOv5l and YOLOR-P6 models, which can give high accuracy output. However, if the deployment environment expects detection of food items using a live feed, it is more suitable to use YOLOv5s or YOLOv5m variants. Therefore, it is highly advisable to establish the trade-off required between accuracy and speed in order to choose the appropriate model for deployment.

Conclusion and Future Work

We aim to train a deep learning–based object detection algorithm to detect the presence of food items that can trigger an allergic reaction. From previous related work, we could not find an image dataset to suffice our needs; therefore, we build our dataset for the same. Firstly, we curate a list of frequently used food items that contain commonly occurring allergens. Based on that, we collect a vast and diverse dataset of food images and annotate bounding boxes to locate the position of the concerned food items within every image. Our final dataset, Allergen30, contains more than 6000 images with 18,000 + annotations spreading across 30 food labels. Owing to their high performance in terms of both accuracy and speed, we train and compare YOLOv5s, YOLOv5m, YOLOv5l, and YOLOR-P6 algorithms on our custom dataset. Results show that all the algorithms perform decently well with reference to accuracy and achieve more than 0.74 mAP. With an mAP of 0.811, YOLOR-P6 performs the best in terms of accuracy, whereas YOLOv5s reports the fastest inference time of 5 ms. Additionally, we discuss the significance of model selection based on the deployment needs:

- Being the first research attempt to make use of deep learning–based computer vision methods to detect the presence of allergen-based food items, this work provides the initial step toward building a robust detection model that can assist people in avoiding possible allergic reactions.
- We hope that the community extends the dataset to accommodate more possible food labels or adds more images of current labels to upgrade the performance of object detection models.
- Moreover, this work also opens an opportunity to improve upon the existing computer vision methods to better learn the otherwise difficult visual cues related to food items.

Acknowledgements The authors thank GAIN (Axencia Galega de Innovación) for supporting this research (grant number IN607A2019/01). We acknowledge the faculty members of Department of Food Processing Technology and Sri Snehashis Guha, PIC Malda Polytechnic, Malda, for their support to conduct this study.

Author Contribution M.M.: conceptualization, methodology, investigation, validation, formal analysis, writing—original draft preparation; T.C.: methodology, investigation, validation, formal analysis, contribution in writing; T.S.: conceptualization, methodology, investigation, validation, formal analysis, writing—original draft preparation; N.B.: methodology, investigation, validation, formal analysis, contribution in writing; S.S.: methodology, investigation, validation, formal analysis and contribution in writing in relevant section; M.R.: data analysis; writing—review and editing; final draft supervision and monitoring; M.A.S.: review and editing, final draft supervision and monitoring. J.M.L.: review and editing, final draft supervision and monitoring. All authors read and approved the final manuscript.

Data Availability All the data used in the manuscript are available in the tables and figures.

Code Availability Not applicable.

Declarations

Ethics Approval Not applicable.

Consent to Participate All authors have given their full consent to participate.

Consent for Publication All authors have given their full consent for publication.

Conflict of Interest Mayank Mishra declares that he has no conflict of interest. Tanmay Sarkar declares that he has no conflict of interest. Tanupriya Choudhury declares that he has no conflict of interest. Nikunj Bansal declares that he has no conflict of interest. Slim Smaoui declares that he has no conflict of interest. Maksim Rebezov declares that he has no conflict of interest. Mohammad Ali Shariati declares that he has no conflict of interest. Jose Manuel Lorenzo declares that he has no conflict of interest.

References

Abrams EM, Sicherer SH (2016) Diagnosis and management of food allergy. CMAJ 188:1087–1093. https://doi.org/10.1503/cmaj.160124

Al-Sarayreh M, M. Reis M, Qi Yan W, Klette R (2018) Detection of red-meat adulteration by deep spectral–spatial features in hyperspectral images. J. Imaging 4

Arora M, Mangipudi P, Dutta MK (2021) Deep learning neural networks for acrylamide identification in potato chips using transfer learning approach. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02867-2

Arslan B, Memiş S, Sönmez EB, Batur OZ (2022) Fine-grained food classification methods on the UEC FOOD-100 Database. IEEE Trans Artif Intell 3:238–243. https://doi.org/10.1109/TAI.2021.3108126

Azizah LM, Umayah SF, Riyadi S, et al (2017) Deep learning implementation using convolutional neural network in mangosteen surface defect detection. In: 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). pp 242–246

B.S. M, Shinde S, Bhavsar K, et al (2018) Non-destructive method to detect artificially ripened banana using hyperspectral sensing and RGB imaging. In: Proc.SPIE

Baenkler H-W (2008) Salicylate intolerance: pathophysiology, clinical spectrum, diagnosis and treatment. Dtsch Arztebl Int 105:137–142. https://doi.org/10.3238/arztebl.2008.0137

Bisgin H, Bera T, Ding H et al (2018) Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles. Sci Rep 8:6532. https://doi.org/10.1038/s41598-018-24926-7

Bochkovskiy A, Wang C-Y, Liao H (2020) YOLOv4: optimal speed and accuracy of object detection

Bossard L, Guillaumin M, Van Gool L (2014) Food-101 – mining discriminative components with random forests BT - computer vision – ECCV 2014. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds). Springer International Publishing, Cham, pp 446–461

Bousquet J, Björkstén B, Bruijnzeel-Koomen CA et al (1998) Scientific criteria and the selection of allergenic foods for product labelling. Allergy 53:3–21. https://doi.org/10.1111/j.1398-9995.1998.tb04987.x

Boye JI (2012) Food allergies in developing and emerging economies: need for comprehensive data on prevalence rates. Clin Transl Allergy 2:25. https://doi.org/10.1186/2045-7022-2-25

Bush RK, Hefle SL (1996) Food allergens. Crit Rev Food Sci Nutr 36:119–163. https://doi.org/10.1080/10408399609527762

Chen C-H, Karvela M, Sohbati M et al (2018) PERSON-Personalized Expert Recommendation System for Optimized Nutrition. IEEE Trans Biomed Circuits Syst 12:151–160. https://doi.org/10.1109/TBCAS.2017.2760504

Chen J, Ngo C (2016) Deep-based ingredient recognition for cooking recipe retrieval. In: Proceedings of the 24th ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, pp 32–41

Chen X, Zhu Y, Zhou H, et al (2017) ChineseFoodNet: a large-scale image dataset for Chinese food recognition

Ciocca G, Napoletano P, Schettini R (2018) CNN-based features for retrieval and classification of food images. Comput vis Image Underst 176–177:70–77. https://doi.org/10.1016/j.cviu.2018.09.001

Ciocca G, Napoletano P, Schettini R (2017a) Food recognition: a new dataset, experiments, and results. IEEE J Biomed Heal Informatics 21:588–598. https://doi.org/10.1109/JBHI.2016.2636441

Ciocca G, Napoletano P, Schettini R (2017b) Learning CNN-based features for retrieval of food images BT - new trends in

image analysis and processing – ICIAP 2017b. In: Battiato S, Farinella GM, Leo M, Gallo G (eds). Springer International Publishing, Cham, pp 426–434

Comas-Basté O, Sánchez-Pérez S, Veciana-Nogués MT, et al (2020) Histamine intolerance: the current state of the art. Biomolecules 10: https://doi.org/10.3390/biom10081181

da Costa AZ, Figueroa HEH, Fracarolli JA (2020) Computer vision based detection of external defects on tomatoes using deep learning. Biosyst Eng 190:131–144. https://doi.org/10.1016/j.biosystemseng.2019.12.003

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. IEEE Comput Soc Conf Comput vis Pattern Recognit (CVP'05) 1:886–893. https://doi.org/10.1109/CVPR.2005.177

Donadello I, Dragoni M (2019) Ontology-driven food category classification in images BT - image analysis and processing – ICIAP 2019. In: Ricci E, Rota Bulò S, Snoek C, et al. (eds). Springer International Publishing, Cham, pp 607–617

Ege T, YANAI K, (2018) Image-based food calorie estimation using recipe information. IEICE Trans Inf Syst E01.D:1333–1341. https://doi.org/10.1587/transinf.2017MVP0027

Fan S, Li J, Zhang Y et al (2020) On line detection of defective apples using computer vision system combined with deep learning methods. J Food Eng 286:110102. https://doi.org/10.1016/j.jfoodeng.2020.110102

Farinella G, Allegra D, Stanco F (2014) A benchmark dataset to study the representation of food images

Farinella GM, Allegra D, Moltisanti M et al (2016) Retrieval and classification of food images. Comput Biol Med 77:23–39. https://doi.org/10.1016/j.compbiomed.2016.07.006

Fu Z, Chen D, Li H (2017a) ChinFood1000: a large benchmark dataset for Chinese food recognition. In: Bevilacqua V, Premaratne P, Gupta P (eds) Intelligent Computing Theories and Application. ICIC 2017a. Lecture Notes in Computer Science. Springer

Fu Z, Chen D, Li H (2017b) ChinFood1000: a large benchmark dataset for Chinese food recognition BT - intelligent computing theories and application. In: Bevilacqua V, Premaratne P, Gupta P (eds) Huang D-S. Springer International Publishing, Cham, pp 273–281

Gallo I, Ria G, Landro N, Grassa RL (2020) Image and text fusion for UPMC Food-101 using BERT and CNNs. In: 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). pp 1–6

Gc S, Saidul MdB, Zhang Y et al (2021) Using deep learning neural network in artificial intelligence technology to classify beef cuts. Front Sensors 2:5. https://doi.org/10.3389/fsens.2021.654357

Geng Z, Shang D, Han Y, Zhong Y (2019) Early warning modeling and analysis based on a deep radial basis function neural network integrating an analytic hierarchy process: a case study for food safety. Food Control 96:329–342. https://doi.org/10.1016/j.foodcont.2018.09.027

Gupta RS, Warren CM, Smith BM et al (2019) Prevalence and severity of food allergies among US adults. JAMA Netw Open 2:e185630. https://doi.org/10.1001/jamanetworkopen.2018.5630

Han Y, Liu Z, Khoshelham K, Bai SH (2021) Quality estimation of nuts using deep learning classification of hyperspectral imagery. Comput Electron Agric 180:105868. https://doi.org/10.1016/j.compag.2020.105868

Hao W, Zhili S (2020) Improved mosaic: algorithms for more complex images. J Phys Conf Ser 1684:12094. https://doi.org/10.1088/1742-6596/1684/1/012094

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 770–778

He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition BT - computer vision – ECCV 2014. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds). Springer International Publishing, Cham, pp 346–361

Hinton G, Deng L, Yu D et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29:82–97. https://doi.org/10.1109/MSP.2012.2205597

Hinton GE, Mcclelland JL, Rumelhart DE (1986) Distributed representations (memory storage). Parallel Distrib Process Explor Microstruct Cogn 77–109

Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu J, Zhao D, Zhang Y et al (2021) Real-time nondestructive fish behavior detecting in mixed polyculture system using deep-learning and low-cost devices. Expert Syst Appl 178:115051

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 2261–2269

Islam SMM, Rahman A, Prasad N, et al (2019) Identity authentication system using a support vector machine (SVM) on radar respiration measurements. In: 2019 93rd ARFTG Microwave Measurement Conference (ARFTG). pp 1–5

Jagtap S, Bhatt C, Thik J, Rahimifard S (2019) Monitoring potato waste in food manufacturing using image processing and internet of things approach. Sustain. 11

Jahani Heravi E, Habibi Aghdam H, Puig D (2018) An optimized convolutional neural network with bottleneck and spatial pyramid pooling layers for classification of foods. Pattern Recognit Lett 105:50–58. https://doi.org/10.1016/j.patrec.2017.12.007

Jia W, Li Y, Qu R et al (2019) Automatic food detection in egocentric images using artificial intelligence technology. Public Health Nutr 22:1168–1179. https://doi.org/10.1017/S1368980018000538

Jiang B, He J, Yang S et al (2019) Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. Artif Intell Agric 1:1–8. https://doi.org/10.1016/j.aiia.2019.02.001

Kaur P, Sikka K, Wang W, et al (2019) FoodX-251: a dataset for fine-grained food classification

Kawano Y, Yanai K (2014) Food image recognition with deep convolutional features. 589–593

Kawano Y, Yanai K (2015) Automatic expansion of a food image dataset leveraging existing categories with domain adaptation BT - computer vision - ECCV 2014 workshops. In: Bronstein MM, Rother C (eds) Agapito L. Springer International Publishing, Cham, pp 3–17

Khaki S, Pham H, Han Y et al (2021) DeepCorn: a semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation. Knowledge-Based Syst 218:106874. https://doi.org/10.1016/j.knosys.2021.106874

Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539

LeCun Y, Boser B, Denker JS et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551

Lee MC, Chiu SY, Chang JW (2017) A deep convolutional neural network based Chinese menu recognition app. Inf Process Lett 128:14–20. https://doi.org/10.1016/j.ipl.2017.07.010

Lin T-Y, Goyal P, Girshick R et al (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42:318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Liu C, Cao Y, Luo Y et al (2018a) A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. IEEE Trans Serv Comput 11:249–261. https://doi.org/10.1109/TSC.2017.2662008

Liu C, Cao Y, Luo Y, et al (2016a) DeepFood: deep learning-based food image recognition for computer-aided dietary assessment

Liu J-H, Sun X, Young JM et al (2018b) Predicting pork loin intramuscular fat using computer vision system. Meat Sci 143:18–23. https://doi.org/10.1016/j.meatsci.2018.03.020

Liu W, Anguelov D, Erhan D, et al (2016b) SSD: single shot MultiBox detector BT - computer vision – ECCV 2016b. In: Leibe B, Matas J, Sebe N, Welling M (eds). Springer International Publishing, Cham, pp 21–37

López-Pedrouso M, Lorenzo JM, Gagaoua M, Franco D (2020) Current trends in proteomic advances for food allergen analysis. Biology 9(9):247. https://doi.org/10.3390/biology9090247

Maintz L, Novak N (2007) Histamine and histamine intolerance. Am J Clin Nutr 85:1185–1196. https://doi.org/10.1093/ajcn/85.5.1185

Mao D, Wang F, Hao Z, Li H (2018) Credit evaluation system based on blockchain for multiple stakeholders in the food supply Chain. Int J Environ Res Public Health 15:1627. https://doi.org/10.3390/ijerph15081627

Martinel N, Foresti GL, Micheloni C (2018) Wide-slice residual networks for food recognition. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp 567–576

McAllister P (2018) Deep learning-based food image classification and crowdsourcing-based calorie estimation approach to support dietary management

McAllister P, Zheng H, Bond R, Moorhead A (2018) Combining deep residual network features with supervised machine learning algorithms to classify diverse food image datasets. Comput Biol Med 95. https://doi.org/10.1016/j.compbiomed.2018.02.008

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133. https://doi.org/10.1007/BF02478259

Mekori YA (1996) Introduction to allergic diseases. Crit Rev Food Sci Nutr 36(Suppl):S1-18. https://doi.org/10.1080/10408399609527756

Mezgec S, Koroušić Seljak B (2017) NutriNet: a deep learning food and drink image recognition system for dietary assessment. Nutrients 9: https://doi.org/10.3390/nu9070657

Mezgec S, Seljak BK (2019) Using deep learning for food and beverage image recognition. In: 2019 IEEE International Conference on Big Data (Big Data). pp 5149–5151

Min W, Liu L, Wang Z, et al (2020) ISIA Food-500: a dataset for large-scale food recognition via stacked global-local attention network

Min W, Wang Z, Liu Y, et al (2021) Large scale visual food recognition

Muthukumar J, Selvasekaran P, Lokanadham M, Chidambaram R (2020) Food and food products associated with food allergy and food intolerance - an overview. Food Res Int 138:109780. https://doi.org/10.1016/j.foodres.2020.109780

Myers A, Johnston N, Rathod V, et al (2015) Im2Calories: towards an automated mobile vision food diary. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp 1233–1241

Naritomi S, Tanno R, Ege T, Yanai K (2018) FoodChangeLens: CNN-based food transformation on HoloLens. In: 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR). pp 197–199

Nasiri A, Omid M, Taheri-Garavand A (2020) An automatic sorting system for unwashed eggs using deep learning. J Food Eng 283:110036. https://doi.org/10.1016/j.jfoodeng.2020.110036

Nowak-Węgrzyn A, Chehade M, Groetch ME et al (2017) International consensus guidelines for the diagnosis and management of food protein-induced enterocolitis syndrome: executive summary-Workgroup Report of the Adverse Reactions to Foods Committee, American Academy of Allergy, Asthma & Immunology. J Allergy Clin Immunol 139:1111-1126.e4. https://doi.org/10.1016/j.jaci.2016.12.966

Nwaru BI, Hickstein L, Panesar SS et al (2014) Prevalence of common food allergies in Europe: a systematic review and meta-analysis. Allergy 69:992–1007. https://doi.org/10.1111/all.12423

Ortolani C, Pastorello EA (2006) Food allergies and food intolerances. Best Pract Res Clin Gastroenterol 20:467–483. https://doi.org/10.1016/j.bpg.2005.11.010

Pandey P, Deepthi A, Mandal B, Puhan NB (2017) FoodNet: recognizing foods using ensemble of deep networks. IEEE Signal Process Lett 24:1758–1762. https://doi.org/10.1109/LSP.2017.2758862

Pereira B, Venter C, Grundy J et al (2005) Prevalence of sensitization to food allergens, reported adverse reaction to foods, food avoidance, and food hypersensitivity among teenagers. J Allergy Clin Immunol 116:884–892. https://doi.org/10.1016/j.jaci.2005.05.047

Pfisterer KJ, Amelard R, Chung AG, Wong A (2018) A new take on measuring relative nutritional density: the feasibility of using a deep neural network to assess commercially-prepared puréed food concentrations. J Food Eng 223:220–235. https://doi.org/10.1016/j.jfoodeng.2017.10.016

Ramos RP, Gomes JS, Prates RM et al (2021) Non-invasive setup for grape maturation classification using deep learning. J Sci Food Agric 101:2042–2051. https://doi.org/10.1002/jsfa.10824

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection

Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger

Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement

Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. MIT Press, Cambridge, MA, USA, pp 91–99

Rich J, Haddadi H, Hospedales T (2016) Towards bottom-up analysis of social food

Rodríguez FJ, García A, Pardo PJ et al (2018) Study and classification of plum varieties using image analysis and deep learning techniques. Prog Artif Intell 7:119–127. https://doi.org/10.1007/s13748-017-0137-1

Rong D, Xie L, Ying Y (2019) Computer vision detection of foreign objects in walnuts using deep learning. Comput Electron Agric 162:1001–1010. https://doi.org/10.1016/j.compag.2019.05.019

Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65:386–408. https://doi.org/10.1037/h0042519

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536. https://doi.org/10.1038/323533a0

Sahoo D, Hao W, Ke S, et al (2019) FoodAI: food image recognition via deep learning for smart food logging. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Association for Computing Machinery, New York, NY, USA, pp 2260–2268

Sergi C, Villanacci V, Carroccio A (2021) Non-celiac wheat sensitivity: rationality and irrationality of a gluten-free diet in individuals affected with non-celiac disease: a review. BMC Gastroenterol 21:5. https://doi.org/10.1186/s12876-020-01568-6

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv 14091556

Singla A, Yuan L, Ebrahimi T (2016) 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. Association for Computing Machinery, New York, NY, USA, pp 3–11

Song Q, Zheng Y-J, Xue Y et al (2017) An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination. Neurocomputing 226:16–22. https://doi.org/10.1016/j.neucom.2016.11.018

Soni A, Al-Sarayreh M, Reis MM, Brightwell G (2021) Hyperspectral imaging and deep learning for quantification of Clostridium sporogenes spores in food products using 1D- convolutional neural networks and random forest model. Food Res Int 110577. https://doi.org/10.1016/j.foodres.2021.110577

Stadelman WJ (2003) EGGS | dietary importance. Encyclopedia of Food Sciences and Nutrition (Second Edition). Academic Press, USA, pp 2009–2012

Sun X, Young J, Liu J-H, Newman D (2018a) Prediction of pork loin quality using online computer vision system and artificial intelligence model. Meat Sci 140:72–77. https://doi.org/10.1016/j.meatsci.2018a.03.005

Sun Y, Wei K, Liu Q, et al (2018b) Classification and discrimination of different fungal diseases of three infection levels on peaches using hyperspectral reflectance imaging analysis. Sensors (Basel) 18. https://doi.org/10.3390/s18041295

Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Szilagyi A, Ishayek N (2018) Lactose intolerance, dairy avoidance, and treatment options. Nutrients 10:1994. https://doi.org/10.3390/nu10121994

Taheri-Garavand A, Nasiri A, Banan A, Zhang Y-D (2020) Smart deep learning-based approach for non-destructive freshness diagnosis of common carp fish. J Food Eng 278:109930. https://doi.org/10.1016/j.jfoodeng.2020.109930

Tan W, Zhao C, Wu H (2016) Intelligent alerting for fruit-melon lesion image based on momentum deep learning. Multimed Tools Appl 75:16741–16761. https://doi.org/10.1007/s11042-015-2940-7

Tatsuma A, Aono M (2016) Food image recognition using covariance of convolutional layer feature maps. IEICE Trans Inf Syst 99-D:1711–1715

Taylor S, Hefle S (2001) Food allergies and other food sensitivities. Food Technol 55

Temple JL, Bernard C, Lipshultz SE et al (2017) The safety of ingested caffeine: a comprehensive review. Front Psychiatry 8:80. https://doi.org/10.3389/fpsyt.2017.00080

Termritthikun C, Muneesawang P, Kanprachar S (2017) NU-InNet: Thai food image recognition using convolutional neural networks on smartphone. J Telecommun Electron Comput Eng 9:63–67

Tompson J, Jain A, LeCun Y, Bregler C (2014) Joint training of a convolutional network and a graphical model for human pose estimation. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1. MIT Press, Cambridge, MA, USA, pp 1799–1807

Viola P, Jones M (2001) "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I. https://doi.org/10.1109/CVPR.2001.990517

Wang C-Y, Liao H-YM, Yeh I-H et al (2020) CSPNet: a new backbone that can enhance learning capability of CNN. IEEE/CVF Conf Comput vis Pattern Recognit Work 2020:1571–1580

Wang C-Y, Yeh I-H, Liao H (2021) You only learn one representation: unified network for multiple tasks

Wang Z, Hu M, Zhai G (2018) Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data. Sensors 18

Widrow B, Hoff ME (1988) Adaptive switching circuits. Neurocomputing: Foundations of Research. MIT Press, Cambridge, MA, USA, pp 123–134

Wu N, Zhang C, Bai X, et al (2018) Discrimination of Chrysanthemum varieties using hyperspectral imaging combined with a deep convolutional neural network. Molecules 23. https://doi.org/10.3390/molecules23112831

Xiao G, Wu Q, Chen H et al (2020) A deep transfer learning solution for food material recognition using electronic scales. IEEE Trans Ind Informatics 16:2290–2300. https://doi.org/10.1109/TII.2019.2931148

Yadav S, Sengar N, Singh A et al (2021) Identification of disease using deep learning and evaluation of bacteriosis in peach leaf. Ecol Inform 61:101247. https://doi.org/10.1016/j.ecoinf.2021.101247

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

Yu X, Lu H, Wu D (2018a) Development of deep learning method for predicting firmness and soluble solid content of postharvest Korla fragrant pear using Vis/NIR hyperspectral reflectance imaging. Postharvest Biol Technol 141:39–49. https://doi.org/10.1016/j.postharvbio.2018a.02.013

Yu X, Tang L, Wu X, Lu H (2018b) Nondestructive freshness discriminating of shrimp using visible/near-infrared hyperspectral imaging technique and deep learning algorithm. Food Anal Methods 11:768–780. https://doi.org/10.1007/s12161-017-1050-8

Yu X, Wang J, Wen S et al (2019) A deep learning based feature extraction method on hyperspectral images for nondestructive prediction of TVB-N content in Pacific white shrimp (Litopenaeus vannamei). Biosyst Eng 178:244–255. https://doi.org/10.1016/j.biosystemseng.2018.11.018

Zaidi SSA, Ansari MS, Aslam A et al (2022) A survey of modern deep learning based object detection models. Digit Signal Process 126:103514. https://doi.org/10.1016/j.dsp.2022.103514

Zhang J, Dai L, Cheng F (2021) Identification of corn seeds with different freezing damage degree based on hyperspectral reflectance imaging and deep learning method. Food Anal Methods 14:389–400. https://doi.org/10.1007/s12161-020-01871-8

Zhang W, Zhang Y, Zhai J et al (2018) Multi-source data fusion using deep learning for smart refrigerators. Comput Ind 95:15–21. https://doi.org/10.1016/j.compind.2017.09.001

Zheng J, Zou L, Wang ZJ (2018) Mid-level deep food part mining for food image recognition. IET Comput vis 12:298–304. https://doi.org/10.1049/iet-cvi.2016.0335

Zhou X, Sun J, Tian Y et al (2020) Development of deep learning method for lead content prediction of lettuce leaf using hyperspectral images. Int J Remote Sens 41:2263–2276. https://doi.org/10.1080/01431161.2019.1685721

Zhou X, Yao C, Wen H, et al (2017) EAST: an efficient and accurate scene text detector