Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 143–152

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Food Image Classification and Data Extraction Using Convolutional Neural Network and Web Crawlers

Chaitanya A[a], Jayashree Shetty[a,*], Priyamvada Chiplunkar[a]

[a]*Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India*

Abstract

Food image recognition is one among the various propitious applications in the area of computer vision. An application with the ability to identify all kinds of food images along with its nutritious value will help people in maintaining a balanced diet. The proposed convolutional neural network (CNN) model can be used for the identification and classification of food images. A pre-trained Inception v3 CNN model is employed via transfer learning to galvanize the original custom-made CNN framework. With the aid of this pre-trained model, the learning process is boosted and is hence more efficient. Data augmentation is performed on the training set as it improves the robustness of the model and as it also helps avoid overfitting. The predicted food label generated by the model is forwarded to the web crawlers for information retrieval. The web crawlers are deployed on the browser through automation for the retrieval of relevant information such as the food item's origin, nutritional details, its recipe, and even the nearby restaurants that serve the dish. The crawlers are built using Python Scrapy and the process of scraping data from the websites is automated through Selenium. The model achieved an accuracy of 97.00% for 20 classes. This model can be further enhanced in terms of scalability by building a deeper more advanced neural network, collecting more images per class for each of the respective food items and by fine-tuning the model hyperparameters.

Keywords: Convolutional neural networks (CNN); Transfer Learning; Inception v3; Food Recognition; Web crawlers; Python; Scrapy; Selenium

1. Introduction

In the past few years, people are increasingly conscious of their health and dilatory intake. Due to the lack of knowledge on nutrition, people still maintain unhealthy lifestyles and bad dietary consumption. It is very essential to know the type of food intake and its corresponding nutritional value [1]. Food image recognition systems have gained more attention to facilitate the recognition and evaluation of food images. It furnishes a way to evaluate the eating habits of the people by estimating the dietary caloric intake [2].

* Corresponding author.
E-mail address: jayashree.sshetty@manipal.edu

1877-0509
10.1016/j.procs.2022.12.410

In recent years, CNN has gained popularity for image categorization. By being a variant of the standard deep neural network (DNN), CNN is made up of alternate convolutional and pooling layers. CNN is advantageous over hard-crafted feature extraction due to its ability to learn optimal features suitably [11] [7]. This study is aimed in use of CNN- aided techniques for food image evaluation and web crawling for information retrieval. Overall, the purpose of the proposed work is to provide a standalone scalable real world model that can categorize food images into different types and provide nutritional values, hence helping the user identify the health benefits or health issues on consumption. The crawler retrieves food origin data along with the recipes which help the user identify key ingredients and their nutritional values [15] and hence help them make smart decisions on whether the particular item is healthy or not for him/her. It also fetches data concerning the availability of a particular food item in nearby restaurants.

2. Literature Review

The authors in [1] have proposed an approach for food classification using a deep CNN approach. The model was created using the data set by collecting images from various sources. The accuracy was tested using various pre-trained models such as AlexNet, Resnet, GoogleNet, K-foodNet, etc, and proved that K-foodNet resulted in the best accuracy.

In [2] the authors went for a more complex approach with the inclusion of NLP. They required the same due to the generic nature of the data scraped from various online repositories. In the proposed approach the targets for data are vetted based on quality and only the most trusted websites are used for scraping, this results in extraction of very specific data without compromising the integrity of the same.

The effective implementation of python's Scrapy for web crawling for data analytics and ingredient/recipe data scraping is seen in [3] and [15]. In [4] the authors implemented a cascaded neural network that improved the accuracy of the AlexNet pre-trained model by 12.28% on the Food-101 data set which is very commendable. The proposed work utilizes a far more advanced pre-trained network in the Inception v3, so a more traditional CNN was able to produce top results, but utilization of a parallel cascaded network could possibly make the proposed model even better while training it for a larger number of classes. In [5] the authors have used the SIFT method for feature extraction. The paper proved that backpropagation neural networks (BPNN) resulted in higher accuracy than k-dimensional trees. In the proposed approach, feature extraction is done automatically using CNN.

The authors in [6] have used the food-11 dataset (dataset with 11 categories of food) and achieved 92.86% accuracy with the aid of the inception V3 pre-trained model. Our proposed approach achieved 97.00% for 20 classes and 96.52% for 25 classes using the Inception V3 pre-trained model. Moreover, the proposed method utilizes web scraping to display necessary details about the particular food item. In [8] and [9] the authors have used HTML parsing for smart online shopping. The proposed framework has used scrapy's selector module for quicker parsing and easier implementation. In [10] the authors used Selenium to automate the scraping process. The proposed Web Scraper model utilizes the same for smooth and efficient web crawling.

In [11] the authors performed classification of Bengali food images, they utilized the VGG16 model for transfer learning resulting in the attainment of 98% for both the F1 score and accuracy. The proposed approach used the inception V3 for transfer learning. The accuracy achieved was 97% and F1 score of 0.99867 proving that it's a more scalable model that produces highly precise results while using nearly thrice the number of classes which makes this proposed model more practical to implement in the real world where need for analysis of vast variety of data is the present-day requirement.

The authors in [13] have created the dataset with 11 classes which contains the images collected from various sources. Authors have also implemented very deep convolutional networks (24 weight layers) and showed that it resulted in greater accuracy. The authors in [16] built a custom 6-layer convolutional network for feature extraction and classification on 20 classes where each class consisted of 500 images. The training and validation accuracy was 93.29% and 78.7% respectively a gap of nearly 15% indicates that the model was overfitting on the data. The proposed model takes care of this by utilizing the power of the inception V3 along with a few custom layers with regularization which play a fine supporting act in the overall framework hence resulting in a max difference of 5% between training and validation accuracy, proving that the model has minimized overfitting to a large extent.

The authors in [18] used a small-scale dataset and checked the effectiveness of fine-tuning and pre-training of CNN. The UECFOOD100 and UECFOOD256 datasets were used to achieve 78.77% and 67.57% accuracy respectively. This project proved the fact that fine-tuned DCNN pre-trained with many categories of images can boost the classification

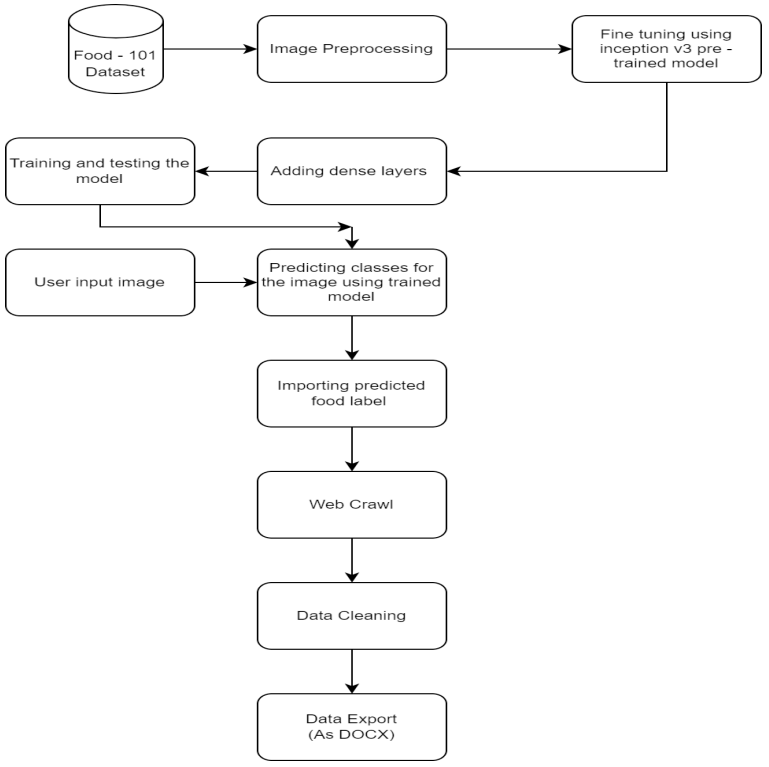

Fig. 1. Steps followed in the proposed approach

performance incredibly. The authors in [20] used the UEC-FOOD100 dataset and got the best accuracy of 70.2%, which is less since the dataset is small. The experiment showed that the color feature is not significant for improving accuracy. Comparably, the proposed approach achieved high accuracy, with pre-trained inception V3 model. It also gives detailed information regarding the particular food item using a web scraper. In [21] the authors depicted the difference in model quality with the aid of ensemble learning and proved that even though the inception V3 is a solid model with accuracy of 93.63% then ensemble model attained an accuracy of 95.54%, the proposed model obtains an accuracy of 97% using inception V3, which implies that proposed model (operating on 20-25 classes) is powerful enough to produce accuracies comparable to ensemble models.

The remainder of this paper is organized as follows: Section 3 explains the methodology of the CNN model and the web scraping technique used. The 4th section explains the results and outcomes obtained. The last section presents the final conclusion.

3. Methodology

The Food-101 dataset comprise of 101 food classes. Every class has 750 training images and 250 manually reviewed test images. The original images with the side length of 512 pixels are scaled to 299 pixels by reducing noise. The images also had wrong labels and intense colors. The proposed approach is depicted in Fig. 1 Some of the random images from a few classes are shown in Fig 2. A random sample function is used to select 20 of these classes for training and testing purposes. This dataset was uploaded to google drive which was then mounted on google colab for use. Colab's Tesla T4 GPU was used for training the CNN model as it is much faster than the conventional CPU. The proposed model is developed and tested in the Python 3.7.13 Jupyter notebook environment due to its ease of use and the availability of a wide variety of ML friendly tools.

Fig. 2. A random image from a few classes

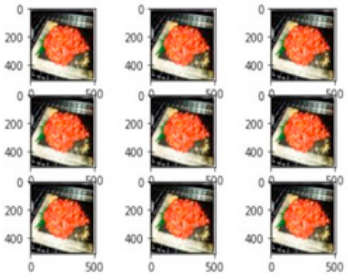

Fig. 3. Different shear ranges of an image

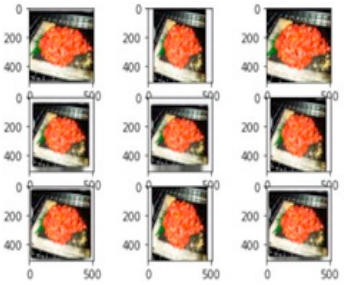

Fig. 4. Different zoom ranges of an image

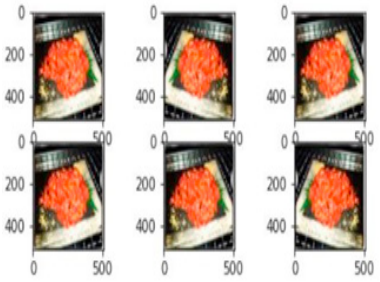

Fig. 5. Depicting horizontal flips of an image

3.1. Data augmentation

Data augmentation is performed on the training set to avoid overfitting and to increase the hardiness of the model. The parameters used are shear range, zoom range, and horizontal flip. The shear range is set to 0.2 and is used to augment the images so that computers can see how humans see things from various angles and as shown in Fig 3. The zoom range is set to 0.2 so that the image will be zoomed by 20% and the result is shown in Fig 4. Horizontal flip is used because it flips rows and columns horizontally as shown in Fig 5 and hence it contributes in building a powerful model.

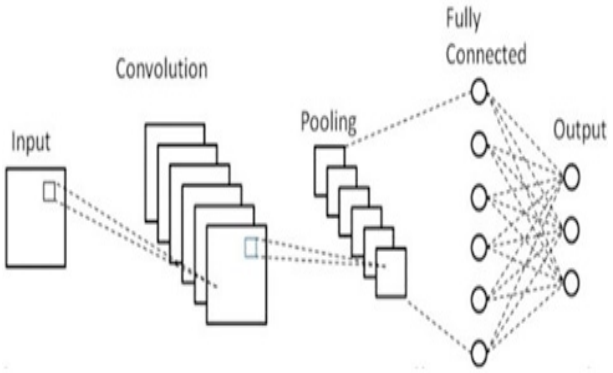

Fig. 6. Types and layers aligned in a CNN model

3.2. Transfer Learning using the Inception V3

A CNN consists of convolutional and pooling layers [1]. It provides an exceptional model architecture for the classification and recognition of the images [5]. The last layer is completely connected, and it represents the output classification of the network. An activity regularizer is applied to this layer to help reduce overfitting, as it penalizes the model in proportion to the magnitude of the activations. This layer hence robustly extracts and learns features from the raw input image to which a sequence of filters was applied. These features will be used by the model for purpose of classification. The different layers used in the model are shown in Fig 6.

The Deep learning library Keras provides various pre-trained models. These deep neural networks are the architectures that are already trained on large datasets. One such model is Inception V3, which is trained on the ImageNet data set [12].

The Inception V3 pre-trained model is fine-tuned using the Food-101 dataset. Using this model, the custom-built layers are appended to the 48 pre-trained layers of Inception V3 to fine-tune the model to the new dataset. Fine- tuning saves more time and computation when compared to the models that are trained from scratch. Table 1 summarizes the fine-tuning applied to the Inception V3.

Table 1. Types and layers aligned in a CNN model

Layer (type)	*Output Shape*	*Param #*
Inception_V3 (Functional)	(None, None, None, 2048)	21802784
conv2d_283 (Conv2D)	(None, None, None, 64)	1179712
max_pooling2d_13 (MaxPooling2D)	(None, None, None, 64)	0
global_average_pooling2d_9 (GlobalAveragePooling2D)	(None, 2048)	0
dense_16 (Dense)	(None, 128)	262272
dropout_9 (Dense)	(None, 128)	0
dense_17 (Dense)	(None, 20)	2580

Total params: 22,993,396
Trainable params: 22,958,964
Non-trainable params: 34,432

Fig. 7 shows the architecture of the inception V3 model. The authors in [20] built a convolutional neural network from the ground up without using any pre-trained model and hence the accuracy attained was a meagre 70.2%, On the contrary, the proposed framework implements the principles of transfer learning resulting in a high precision model. This shows that by powering the model through transfer learning, the model can learn quickly, efficiently and provide highly accurate predictions.

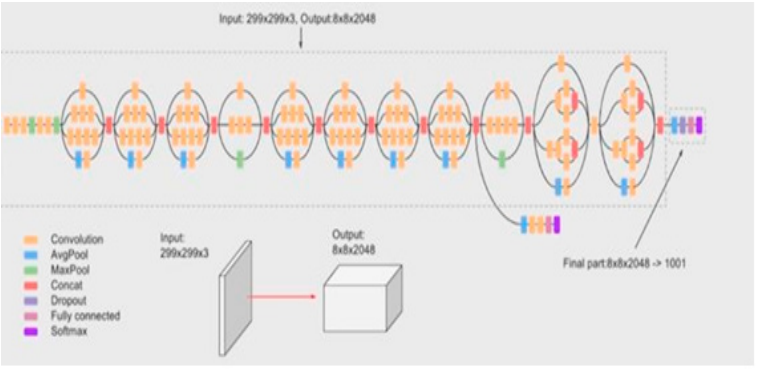

Fig. 7. Architecture of inception V3 model

3.3. Classification

The model is trained for a total of 20 epochs, TensorFlow's SGD optimizer is used for training with its momentum set at 0.9. The optimum value of the learning rate found here is 0.0001. The categorical cross-entropy loss function is deployed, as the proposed model is required to perform multi-class image classification.

3.4. Web Scraper

Python Scrapy has the added benefit of providing methods to clean the data extracted by designing containers known as Items, which provides a wide variety of data manipulation facilities and it possesses pipelining features as well, with the help of which extracting and exporting various types of multimedia resources from the internet straightforward. Scrapy Spiders within a project file are auto-assigned some settings concerning the scraping task at hand, which can be configured to the required needs, thereby making it safe to scrape the websites while also not putting much burden on the website servers.The food label is imported from the image recognition model into the main driver code and will be further forwarded/imported into 3 different Scrapy Spider classes.

- Restaurant Spider: is used to scrape restaurant data from www.Zomato.com with real-time location service enabled.
- Recipe/Nutrition Spider: is used to scrape ingredients, recipes, and nutritional information from the www.Allrecipes.com website.
- Wiki Spider: is used to scrape origin and course information from www.Wikipedia.org.

Choosing the most reliable websites is the recipe for success for any competent web scraper, as it greatly reduces the workload and latency of the scraping process and ensures the integrity of the data scraped.All the spiders crawl their respective sites with the help of the tools provided by Scrapy. The spiders navigate to the requested information using python's Selenium's web driver [10] and once the required information is located the data is parsed/scraped from the HTML tree using Scrapy's Selector class, which utilizes its XPath module.Selenium is a simple automation tool whose only prerequisite is the web browser's device driver; it creates a temporary test environment to carry out the scraping and it can handle both HTML and JavaScript pages. With the help of the web driver's options features, it's possible to activate geolocation services, hence the browser will automatically detect your location without any prompting, its speed is reliant on the local internet connectivity and it provides a lot of browser options which makes the spider more website friendly and hence harder to get blocked from the website. A selector is a simple tool by which data can be easily extracted from the page source of the web page that the selenium driver is currently on. It provides many alternatives for data extraction while scraping.

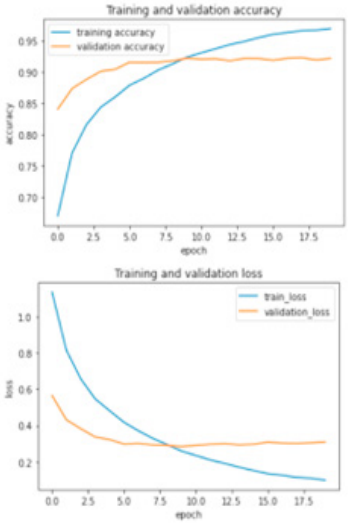

Fig. 8. Training and validation accuracies and losses for 20 classes of Food-101

3.5. Data Cleaning and export

The data scraped is loaded into the main Spider code which handles the data cleaning and integration, through list slicing. It utilizes the panda's library for data frame generation to increase the readability of the scraped data and to constitute for a user-friendly visualization. Finally, the cleaned data is ready to be presented to the user. The code contains a utility function to neatly append the data to a word document in real-time. The final document contains all the scraped information arranged neatly for ease of visualization and comprehension. To avoid getting blocked while web scraping, the download delay of 3-5 secs between each of the pages scraped is maintained. To prevent spider bot detection, scrape at off-peak website traffic hours and finally disable cookies of the website which is intended to be scraped. The web driver's headless option is can be utilized to make the automated browsers invisible.

4. Results and Discussion

The model is very precise as it provides high accuracy of 97%. The web spiders are rapid, and easily configurable with various options and settings to make the entire process very efficient and streamlined as well as secure while scraping from the internet. Initially for the model, 20 classes were selected through random function and achieved 97.00% training accuracy and 92.23% validation accuracy for 20 epochs. Training loss is 0.10 and validation loss is 0.3092. When the number of classes is increased to 25, the training accuracy is 96.52% and validation accuracy is 91.46% with the losses of 0.1206 and 0.3576 respectively, this shows that the model is scalable to an extent without much compromise in accuracy. The time taken for each image is 1.6 seconds, so the total time taken for 20 epochs is 2.27 hours on Google Colab GPU. The graph in Fig. 8 shows the accuracies and losses of the model with 20 classes. It is observed that the gap between training and validation accuracy in the graph decreased as the number of epochs increased and the final difference is at approximately 5%, this proves that there is minimal over-fitting of the data.

The confusion matrix is evaluated to estimate the effectiveness of the model. The confusion matrix for the result obtained is shown in the below Fig. 9. This gives the values of true positive, true negative, false positive, and false-negative results. Hence, the accuracy can be easily found using the formula given by Eq. 1

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

For a good model, there should be a low error rate. It can be evaluated using Eq. 2

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

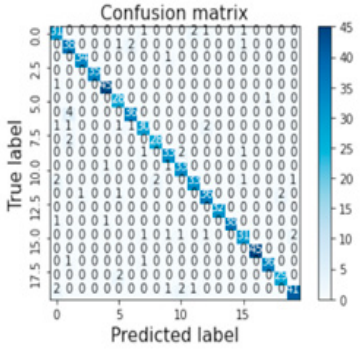

Fig. 9. Confusion matrix for 20 classes

	precision	recall	f1-score	support
apple_pie	1.00	1.00	1.00	37
caesar_salad	1.00	1.00	1.00	38
chicken_wings	1.00	1.00	1.00	49
edamame	1.00	1.00	1.00	41
french_fries	0.98	1.00	0.99	42
fried_rice	1.00	1.00	1.00	34
greek_salad	1.00	1.00	1.00	47
grilled_salmon	1.00	1.00	1.00	37
guacamole	1.00	1.00	1.00	36
hamburger	1.00	0.97	0.99	34
hot_dog	1.00	1.00	1.00	33
ice_cream	1.00	1.00	1.00	43
lasagna	1.00	1.00	1.00	38
lobster_bisque	0.98	1.00	0.99	46
onion_rings	1.00	1.00	1.00	35
pancakes	1.00	1.00	1.00	33
pho	1.00	0.97	0.98	32
seaweed_salad	1.00	1.00	1.00	42
spaghetti_bolognese	1.00	1.00	1.00	32
strawberry_shortcake	1.00	1.00	1.00	21

Fig. 10. Class-wise performance of the fine-tuned Inception V3

It can be noticed in the figure that this model is very effective since the accuracy is high and the error rate is low. The Precision, Recall and F1 values returned by the model using the sklearn.metric library have all achieved a remarkably high score of 0.99867. Precision defines the proportion of correct observations among the positively classified observations. The precision metric is obtained with the help of the following Eq. 3

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall metric gives us the estimate of how many positive observations were classified correctly. This metric can be computed with the help of the following Eq. 4

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1 measure is the harmonic mean of Precision and Recall metrics that we have computed above. F1 score can be calculated using the following Eq. 5

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5}$$

Fig. 10 gives the classification report of the model for the 20 classes, it shows that the model is strong overall and that there is a strong distinction between the various classes. This proves that the proposed model is finely tuned and highly precise.

Fig. 11 depicts the near perfect accuracy score for different classes taken from the sample of 10 from the original set of classes. This shows that the model is unbiased and gives a highly accurate reading of 99.99% for the vast majority

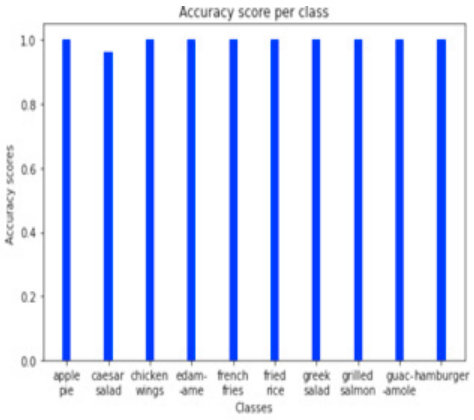

Fig. 11. Accuracy Score per Class

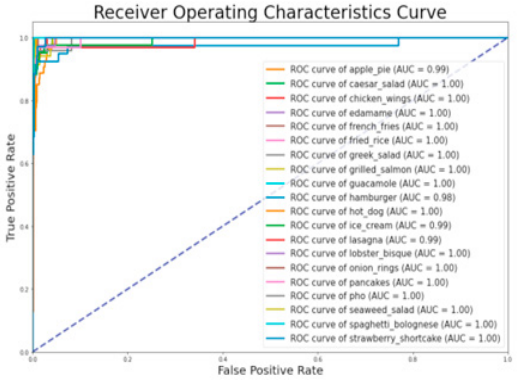

Fig. 12. ROC curve of 20 classes

of classes. Classification performance is also measured by the receiver operating characteristic curve (ROC). The area under the curve (AUC) determines the classification performance of the model. The graph obtained for this model is shown in Fig. 12. As observed, the ROC curves for all the 20 classes have high AUC which implies that the model is highly accurate.

5. Conclusion and Future Work

Image classification is a challenging task because each image possesses certain latent features which makes it unique with respect to the others. The Convolutional neural network approach that has been employed in the proposed framework works excellently and efficiently only when there is a dataset large enough to train the model. To train the model with such a gigantic dataset, it may take hours or even days which is the reason behind the utilization of transfer learning in this approach. The goal is to sustain/improve the model accuracy for a larger number of classes so that it covers almost all the main cuisines consumed all over the globe. It is more than feasible to make further inroads with the help of a more densely connected CNN [19] which can learn the features more quickly and efficiently than it does at present. The pre-trained model (inception V3 in this case) can be tuned to near perfection with the help of AI as seen in [14] which can help achieve a near ideal framework for image classification. Similarly, more advanced web crawling techniques can be implemented to reduce the latency in information retrieval and also expand the search to more sites for crawling as an alternate/backup to ensure data availability at all times. In the future, the work can be expanded to develop a mobile application which makes use of the phone camera with these improved

set of features[17],which would make it a very convenient one stop solution for users to keep track of their diet and thereby maintain their overall health.

References

[1] Park, Seon-Joo, Akmaljon Palvanov, Chang-Ho Lee, Nanoom Jeong, Young-Im Cho and Hae-Jung Lee (2019) "The development of food image detection and recognition model of Korean food for mobile dietary management." *Nutrition research and practice* **13** (6): 521–528.

[2] Yunus, Raza, Omar Arif, Hammad Afzal, Muhammad Faisal Amjad, Haider Abbas, Hira Noor Bokhari, Syeda Nawaz (2018) "A framework to estimate the nutritional value of food in real time using deep learning techniques." *IEEE Access* (7): 2643–2652.

[3] Thomas, David, Mathew, Sandeep Mathur (2019) "Data analysis by web scraping using python" in *3rd IEEE International conference on Electronics, Communication and Aerospace Technology (ICECA)*: 450–454.

[4] Sun, Enji (2021) "Small-scale image recognition based on Cascaded Convolutional Neural Network" in *IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*: 2737–2741.

[5] Giovany, Stanley, Andre Putra, Agus S Hariawan, Lili A Wulandhari (2017) "Machine learning and sift approach for Indonesian food image recognition." *Procedia computer science* **116**: 612–620.

[6] Islam, Md Tohidul, B.M. Nafiz Karim Siddique, Sagidur Rahman, Taskeed Jabid (2018) "Food image classification with convolutional neural network" in *IEEE International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)* (3): 257–262.

[7] Kagaya, Hokuto, Kiyoharu Aizawa, Makoto Ogawa (2014) "Food detection and recognition using convolutional neural network" in *Proceedings of the 22nd ACM international conference on Multimedia*: 1085–1088.

[8] Mahto, Deepak Kumar, Lisha Singh (2016) "A dive into Web Scraper world" in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*: 689–693.

[9] Mehak, Shakra, Rabia Zafar, Sharaz Aslam, Sohail Masood Bhatti (2019) "Exploiting filtering approach with web scrapping for smart online shopping: Penny wise: A wise tool for online shopping" in *2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*: 1–5.

[10] Murali, Ranjani (2018) "An intelligent web spider for online e-commerce data extraction" in *Second International Conference on Green Computing and Internet of Things (ICGCIoT)*: 332–339.

[11] Uddin, Asif Mahbub, Abdullah Al Miraj, Moumita Sen Sarma, Avishek Das, Md Manjurul Gani (2021) "Traditional Bengali Food Classification Using Convolutional Neural Network" in *IEEE Region 10 Symposium (TENSYMP)*: 1–8.

[12] Shen, Yin, Yanxin Yin, Chunjian Zhao, Bin Li, Jun Wang, Guanglin Li, Ziqiang Zhang (2019) "Image recognition method based on an improved convolutional neural network to detect impurities in wheat" *IEEE Access* (7): 162206–162218.

[13] Subhi, Mohammed A., Sawal Md. Ali (2018) "A deep convolutional neural network for food detection and recognition" in *IEEE-EMBS conference on biomedical engineering and sciences (IECBES)*: 284–287.

[14] Tian, Youhui (2020) "Artificial intelligence image recognition method based on convolutional neural network algorithm" *IEEE Access* (8): 125731– 125744.

[15] Chaudhari, Shilpa, R. Aparna, Vinay G Tekkur, GL. Pavan, Shreekanth R Karki (2020) "Ingredient/recipe algorithm using web mining and web scraping for smart chef" in *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*: 1–4.

[16] Reddy, V. Hemalatha, Soumya Kumari, Vinita Muralidharan, Karan Gigoo, Bhushan S. Thakare (2019) "Food Recognition and Calorie Measurement using Image Processing and Convolutional Neural Network" in *4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*: 109–115.

[17] Tomescu, Vlad-Loan (2020) "FoRConvD: An approach for food recognition on mobile devices using convolutional neural networks and depth maps" in *IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI)*: 000129–000134.

[18] Yanai, Keiji, Yoshiyuki Kawano (2015) "Food image recognition using deep convolutional network with pre-training and fine-tuning" in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*: 1–6.

[19] Metwalli, Al-Selvi, Wei Shen and Chase Q. Wu (2020) "Food Image Recognition Based on Densely Connected Convolutional Neural Networks" in *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*: 27–32.

[20] Zhang, Weishan, Dehai Zhao, Wenjuan Gong, Zhongwei Li, Qinghao Lu, Su Yang (2015) "Food image recognition with convolutional neural networks" in *IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing*: 690–693.

[21] Fakhrou, Abdulnaser, Jayakanth Kunhoth, Somaya Al Maadeed (2021) "Smartphone-based food recognition system using multiple deep CNN models" *Multimedia Tools & Applications* **80** 33011—33032.