FoodieCal: A Convolutional Neural Network Based Food Detection and Calorie Estimation System

Shahriar Ahmed Ayon
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Shahriar131519@gmail.com

Chowdhury Zerif Mashrafi
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Zerifmashrafi@gmail.com

Abir Bin Yousuf
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Abiryousuf245@gmail.com

Fahad Hossain
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Fahadc6224@gmail.com

Muhammad Iqbal Hossain
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Iqbal.hossain@bracu.ac.bd

2021 National Computing Colleges Conference (NCCC) | 978-1-7281-6719-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/NCCC49330.2021.9428820

***Abstract*—According to recent studies across the world, we can see that a healthy diet is the key to having a sound health and body. People nowadays are more concerned with their diets than ever before. With the advancement of science, it is now viable to construct a unique food identification system for keeping track of day to day calorie intake. However, building this kind of system creates several complications on constructing and implementing the model. In our paper, we have developed a new neural network based model which will predict the food items from a given image and show us the estimated calorie of the detected food items. In order to achieve our goal, we have prepared a dataset of around 23000 images for 23 different food categories. For this, we have built a system which can detect multiple foods by training CNN with features extracted by Inception V3. We have achieved 89.48% accuracy for this model and we deployed our system on a webpage. The user has to upload an image of food item in the webpage and our system will predict the food item along with the estimated calories in real time.**

***Index Terms*—Food Detection; CNN; ResNet; Inception V3.**

I. Introduction

With the advent of technologies, nowadays, people have become more aware of what they are consuming to satisfy their hunger. This is due to the fact that obesity has become a common phenomenon in recent times. The World Health Organization (WHO) has reported [11] that, in 2016, around 1.9 billion people (aged 18 years and above) around the world are suffering from overweight and 650 million among them are obese. 40 million children (under the age of 5) were overweight in 2018 which is an alarming sign for the world. In 2019, an estimated 38.2 million children under the age of 5 years were overweight or obese. The recent cases of obesity among young generation is very high. Due to the influence of western lifestyle, the internet and advertisements on TV, the youth including children are more likely to consume fast food and other junk foods more than before which lead to a variety of chronic diseases like heart blockage, stroke, diabetes, liver problems, kidney damage etc. [7] [3]. To avoid these types of problems, food journaling has become popular to help people keep track of how much calorie they should consume in a particular time. So there comes the concept of food detection and calorie estimation using the smartphones or camera and journaling in a digital way. But it presents a lot of uncertainty because there are lots of variations of different food images of different cuisines around the world. The scientists have done years of research and provided us with various types of hypotheses on this issue. Most of the existing systems have a low accuracy in detecting food items let alone showing overall estimated calorie.

This has motivated us to step up and do an extensive research on this issue. We have developed a system that not only identifies the food items with a great accuracy from an image but also shows the amount of calorie the food item contains. In this paper, we have proposed a model which is focused on Convolutional Neural Network (CNN) for food detection from a given image. Initially, we have developed a dataset of 23 food categories for identifying multiple food items. Then we extracted features from the images using Inception_V3 and trained the model with CNN. Next, we deployed the model in a webpage which is for showing the result. In the end, the result is evaluated in the webpage when an image is given as an input and the system predicts what food it is along with showing the calorie amount.

II. Literature Review

Quite a few works have been done on the food recognition process using various types of techniques. But each one of them have their own limitations. We have gone through some of the earlier proposed models on Food identification and investigated them in detail. In the research paper by Joutou et al. 2009 [1] and Hoashi et al. 2010 [2] color, surface, slope and Filter highlights are extracted from food pictures and a partitioned classifier is prepared for each include. At last, all the classifiers are weighted combined with the different parts

of algorithms. Through their developed system 61.3% and 62.5% accuracy is accomplished for 50 and 85 categories of Japanese foods utilizing 9 and 17 highlights (5-fold cross approval in 8500 pictures). This system is additionally giving lower precision in terms of distinguishing food items from a picture. The dataset contains an add-up to a picture of 8500 food items.

In Subhi et al. 2018 research paper [9], the authors proposed a model which can identify food items based on CNN algorithm. They also used a new dataset for local Malaysian food which contains eleven food categories with (5800) images. They utilized two datasets for their proposed model. For food/non-food classification, they have used FOOD-101 dataset and for the grouping of food items, they have applied their proposed local Malaysian food dataset. Additionally, they claimed that they had performed very extensive convolutional networks (24 weight layers) for food image classification. They proposed a more compound model containing of multi convolutional layers blocks before the final fully connected layer. The results confirm the significance of network depth in training visual representations.

III. Proposed Method

A. Dataset Collection

For constructing the dataset for multiple food detector, we used around 23000 images from 23 different food classes. We made sure that all the food classes had the correct images in them before training our algorithm. Both Training Dataset and Validation dataset were thoroughly checked in our dataset. The 23 food classes we used for our dataset are:

- Chicken Wings, Chocolate Cake, Churros, Cupcakes, Deviled Eggs
- Donuts, Fried Rice, Grilled Cheese Sandwich, Hamburger, Hotdog
- Hot and Sour Soup, Ice cream, Lobster Bisque, Macarons, Oysters, Pancakes
- Pizza, Samosa, Seaweed Salad, Steak, Waffles, Tacos, Tuna Tartare

B. Dataset Sample

Fig. 1. Sample Images from the Dataset

We have used total 23 different food categories for making both of our datasets and each food class contains 1000 images. As a result, it is very difficult to show a good sample from the database. We tried to show at least 80 to 90 images from our dataset and we managed to put together 90 images in total in Figure 1. This image given below contains at least 3 to 4 images from each food category and it is visible from this sample image that we used various types and calories of food for constructing our dataset.

C. Data Preprocessing

For the dataset we used for detecting multiple items from an image, we had to follow the steps which are: importing some necessary libraries, initializing directory and resizing the images before extracting features. We imported os, tensorflow, numpy as np, glob, matplotlib.pyplot as plt etc. libraries to start data preprocessing. Again, after importing these libraries, we initialized the directory of our dataset so that the model can get the images and start resizing them and after that start extracting features. We uploaded our whole dataset in the Google Drive and we had a folder named Multiple Food Detector there. In that folder, there were 2 folders named Train and Validation. Lastly, our images were reshaped as Image height=299, Image Width= 299 and Number of Classes = 23. After resizing the images, we proceeded to extract features using our inception v3 model.

D. Convolutional Neural Network

CNN which is known as convolutional neural network. It mainly uses a pooling in order to reduce the processing time or power consumption required to process the data by reducing the dimension [8]. In addition, CNN works in multiple layers where the output of first layer will work as an input for the second layer and the output of the second layer will work as an input for the third layer and so on. After making the images suitable for multilayer perceptron CNN now flattens the image into column vectors. This flatten images are put into a neural network which we call feed forward network and then we use back propagation for backtracking for analysing and finding errors,the process is repeated in every iteration while training[8] [10]. After calculating the error back propagation traverses back to hidden layer or inner layer for adjusting the wight in order to decrease the error. Back propagation keeps repeating this process until the algorithm reaches the potential output[12].

E. Feature Extraction Technique

Inception V3: Inception V3 was first introduced in a research paper developed by Szegedy et al.[6]. It was developed with a goal to modifying the earlier version of Inception architectures by reduceing the computational complexity. In several biomedical applications GoogLeNet [4] achieved tremendous classification performance and Inception V3 turns out to be an extended version of this network. It is a 48 layered network architecture. In terms of computational efficiency,

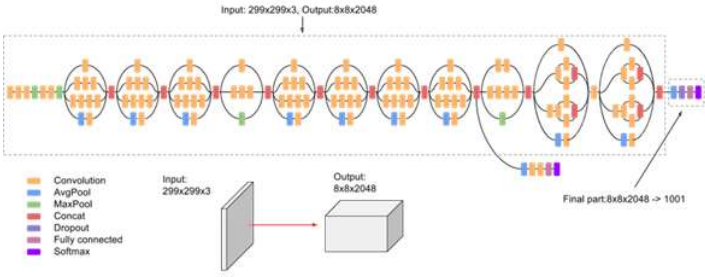

Fig. 2. Inception V3 (Complete Architecture).Recreated from [6]

Inception Networks have outrun the VGGNet. A change is made in an Inception Network only when it is confirmed that computational advantages are not lost. Several techniques including factorized convolutions, regularization, dimension reduction etc. are applied to optimize the network and in an Inception V3 model. This results in losing the constraints and obtaining an easier model.

Residual Network (ResNet): Residual Network (ResNet) is a type of deep convolutional network which was first introduced by He et al. (2016) [5]. The concept of Residual Network (ResNet) was introduced to train and test thousands of convolutional layers with a strong performance. The skip connection basically figures out which layers degrade the accuracy and skip those in training. The authors [5] have allowed the network to learn residual mapping instead of letting the network learn underlying mapping. The residual blocks consist of layers. For example, ResNet-50 has 50 layers of residual blocks. In the proposed residual network model, the authors have denoted the underlying mapping as H(x) and let the network fit residual mapping which gives H(x) := f(x) - x. Finally, the original mapping is casted into H(x) := f(x) + x. Here, F(x) is denoted as the residual function. The authors [1] have claimed that it is easier to get to the solution of optimization of residual mapping F(x) than optimizing the underlying mapping H(x).

IV. Model Description

The whole workflow of our work can be described using a work-flow diagram. From this work-flow diagram shown in Figure 3, we can provide an overview of our work and all of the steps we followed to build our system. These are the major procedures of our code and even if some minor procedures are not shown here, they are included inside these major procedures.

Constructing Dataset: Firstly, when we collected around 23000 images for 23 classes in our dataset and again divided them into Training Dataset and Validation Dataset as before. We used exactly 18413 images for Training Dataset and exactly 4538 images for Validation Dataset.

Data Preprocessing: In order to make a dataset compatible with the system, some steps need to be followed. So, we had to import necessary libraries, initialize directory, resize all the images in our dataset before extracting features.

Building Inception V3 Model: After finishing data preprocessing, we focused on building the inception v3 model which will be used to store the extracted features from the images.

Extracting Features: After building the inception v3 model, we used the model to assign individual weight to all the images in our dataset and these weights were used to train the CNN. To be more specific, we tuned the weights of the last dense layer of inception V3 according to our dataset and these weights were saved in a file named inception model.hdf5.

Training CNN: After finalizing required models, we proceeded to train the CNN (Convolutional Neural Network) according to the inception V3 model. We kept Epochs = 12, Batch Size= 16, Train sample = 16046 and Validation Sample= 3959.

Save Model: We saved our model after the training was done and then started working on our webpage. We saved all of our training, visualize and inception v3 code in the train folder.

Designing Webpage: We designed our webpage as our requirement. We wanted our system to take an image as input and give us the food name and food calorie as output. So we kept an upload image portion in our webpage and then we set the output to be shown right beside the uploaded image.

Deploying Model into the Webpage: After designing the webpage, we imported necessary libraries and we also had to initialize the food categories and the respective food calories for those categories. Then we loaded our trained model in the deployment code.

Uploading Images: After deploying the model, we collected some test images to see if our model is predicting right or not. Then we started uploading images into our webpage and started analyzing the output.

Analyzing Outputs: After uploading an image, we have to click a button named predict and the webpage showed us the name of the food/foods that were predicted from the uploaded image along with the calorie count and prediction accuracy. We also wrote the code in such way so that we get a table

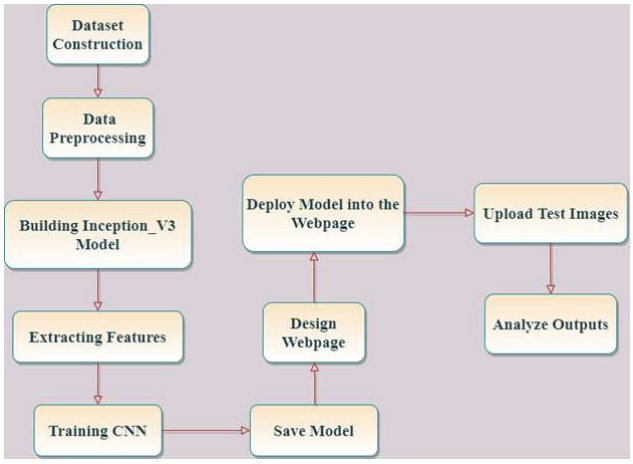

Fig. 3. Work-Flow Diagram of the Proposed System

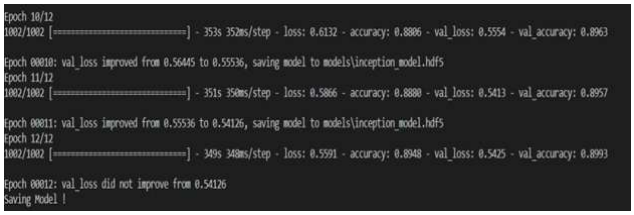

Fig. 4. Train loss, Train accuracy, Validation loss, Validation accuracy

of prediction chart for every image we upload. This chart is shown in the command prompt.

V. Experimentation and Result Analysis

Our main focus is on detecting multiple food in an image. For this we build a model using inception v3 model to get higher accuracy for multiple food image. The model extracted features from all the images in the dataset and saved them as weights. Then we used these saved weights in the inception v3 model to train our CNN.

Firstly, we initialized our inception V3 model and then we used the dense layer for class prediction. We again used a dropout function to prevent our model from overfitting. We used ReLu activation function for both of our dense layers and finally, for predicting the output, we used sigmoid activation function for our last layer. We used loss= categorical crossentrophy, learning rate=0.0001, matrics=accuracy and momentum=0.9 etc. This time we have used 12 epochs and batch size=16.We got the following result shown in Figure 4.

From figure: 4 we can see that for 12 epochs we got 89.48% accuracy and 89.93% validation accuracy. Moreover, as the epoch count goes up, accuracy of our training also goes up. As we put fewer images for validation dataset, the change in validation accuracy and validation loss are not as consistent as training accuracy and training loss. We can show the changes in loss and accuracy with respect to epochs as these following 2 graphs in Figure 05 and Figure 06:

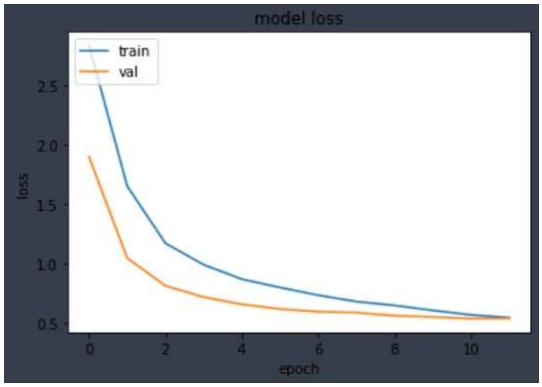

Fig. 5. Changes in loss with respect to epochs

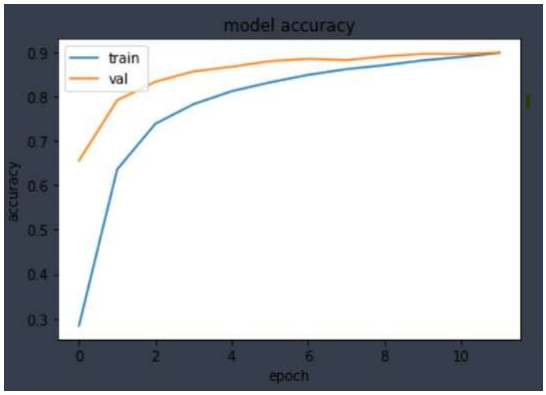

Fig. 6. Changes in accuracy with respect to epochs

A. Analyzing Multiple Food Item from an Image

After designing our webpage, we started to test out system with some sample images. We chose an image using the Choose button and uploaded an image in our website . After uploading, our webpage looked like this shown in the figure 7.

After uploading this image, we clicked the Predict button and the output was shown like the image given in the figure 8.

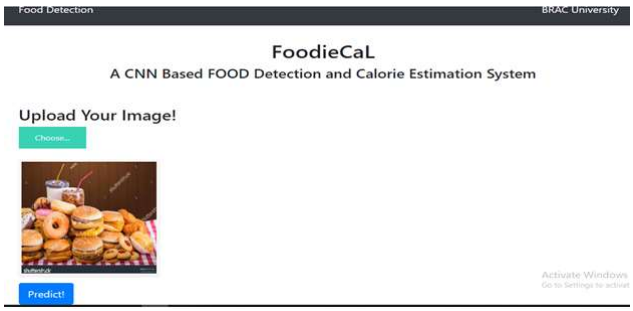

Fig. 7. Test image

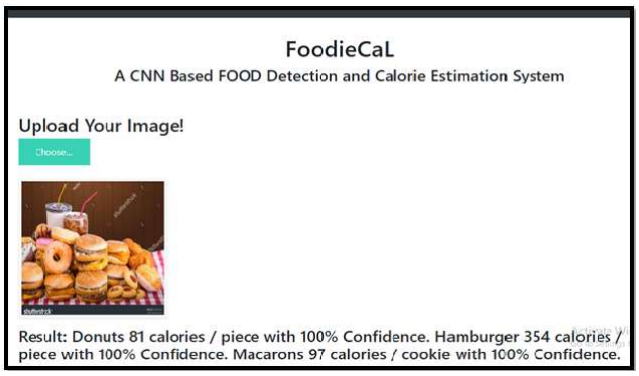

Fig. 8. Output for test image

TABLE I
PREDICTION PERCENTAGE FOR TEST IMAGE 3

Food Category	Prediction Percentage
Chicken Wings	25.10%
Chocolate Cake	24.65%
Churros	89.12%
Cup Cakes	88.44%
Deviled Eggs	54.64%
Donuts	**99.89%**
Fried Rice	25.06%
Grilled Cheese Sandwich	92.29%
Hamburger	**99.69%**
Hot and Sour Soup	09.47%
Hotdog	93.66%
Ice Cream	56.85%
Lobster Bisque	41.54%
Macarons	**99.73%**
Oysters	28.03%
Pancakes	95.17%
Pizza	07.99%
Samosa	23.62%
Seaweed Salad	01.89%
Steak	11.76%
Sushi	78.32%
Tacos	10.68%
Tuna Tartare	30.79%
Waffles	82.20%

The test image shown in Figure 7 and Figure 8 contained Hamburgers, Donuts, Macarons and Milkshakes. We had hamburgers, donuts and macarons in our dataset and our CNN was trained to predict these items from an image. As we did not have milkshakes in our dataset, our system did not detect milkshake. As we can see from the output, our system successfully predicted that the image contained hamburgers, donuts and macarons as well. Our system also printed the calories for these food items and also showed the prediction accuracy as output. We also got a probability chart for this image for all the 23 categories from our dataset. The probability chart was shown in the command prompt as shown in TABLE: 1.

From the image shown in figure 8, our system predicted:

- probability of being Hamburger is 99.69%
- probability of being Donuts is 99.88%
- probability of being Macarons is 99.72%

The TABLE 1 is showing the probability of all our food items after detecting the input image and it is only printing the outputs where prediction percentage is above 96%. To be more specific, we kept the threshold of our prediction percentage at 96 and any food item below that accuracy will not be shown as output. Here Hamburger, Donuts and Macarons are detected because their prediction percentage are above 96%.

B. Analyzing Top Food Items:

For this section, we randomly chose 5 food items and 4 test images shown in Figure 9 which have these selected food items and we analyzed the prediction accuracy for 5 food classes.(Table II)

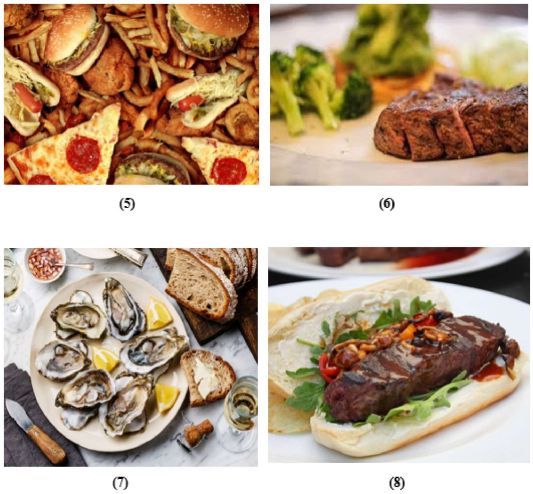

Fig. 9. More Test images

TABLE II
OUTPUT COMPARISON FOR SOME TEST IMAGES

Test Image No.	Pizza	Hamburger	Hotdog	Steak	Oysters
5	97.37%	98.02%	89.24%	44.69%	86.08%
6	71.75%	62.25%	19.30%	99.98%	49.06%
7	50.77%	88.25%	24.02%	48.80%	99.88%
8	58.94%	97.74%	99.72%	97.89%	36.18%

VI. CONCLUSION AND FUTURE WORK

Our main purpose of this study is to identify the calorie of the food from a given image which will help people overcome diseases like obesity, diabetes, heart problem, kidney failure etc. diseases caused by overweight. We believe that if people know about the calories of the food, it will help them to lead a healthier life by keeping track of how much calories they are consuming. So, we have researched and also looked into various methods for the food recognition process. CNN is most suitable for image or object detection processes. In case of performance, CNN outperforms other neural networks and machine learning and deep learning algorithms on conventional 2D or 3D image recognition tasks and other object detection tasks. In terms of statistical results, they also have high calculating efficiency in terms of object or image classification system.

At this stage of our research, we focused on building a Webpage based on CNN algorithm using ResNet and inception v3 models, which is able to classify different types of foods and show us the nutrition value of these foods. We found out a few limitations in our system and they are: Similar looking food items are problematic for multiple food detector because sometimes our system detects multiple food from an image with a single food item. Lastly, the angle at which the image is taken is very important for our system to detect food items accurately. Usually, our system works poorly for the images with top view.

In the upcoming stages, we are planning on overcoming these issues and developing a mobile app that not only identifies the food items with a great accuracy from an image captured on the smartphone but also it will review the medical reports of the user to suggest whether the amount of calorie should be taken or not. Moreover, in the near future we want to work with a bigger dataset for example: a dataset of 100 or more Bengali food items using this model we have shown in this paper. As such a dataset is not available, we are thinking of making our own dataset with 2000 or more images for each food category.

REFERENCES

[1] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2009, pp. 285–288.

[2] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *2010 IEEE International Symposium on Multimedia*, IEEE, 2010, pp. 296–301.

[3] E. J. Gallagher and D. LeRoith, "Obesity and diabetes: The increased risk of cancer and cancer-related mortality," *Physiological reviews*, vol. 95, no. 3, pp. 727–748, 2015.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[7] N. C. Institute, *Obesity and cancer risk*, https://www.cancer.gov/about-cancer/causes-prevention/risk/obesity/obesity-fact-sheet, 2017.

[8] S. Saha, *A comprehensive guide to convolutional neural networks-the eli5 way*, Dec. 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[9] M. A. Subhi and S. M. Ali, "A deep convolutional neural network for food detection and recognition," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, IEEE, 2018, pp. 284–287.

[10] R. Balsys, *Convolutional neural networks (cnn) explained step by step*, Feb. 2020. [Online]. Available: https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-explained-step-by-step-69137a54e5e7.

[11] W. H. Organization, *Obesity and overweight*, https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight, 2020.

[12] *Back propagation neural network: Explained with simple example*. [Online]. Available: https://www.guru99.com/backpropogation-neural-network.html.