ISSN 2277-2685
IJESR/Jan. 2022/ Vol-12/Issue-1/1-15
Kadari Neeraja *et. al.,* / International Journal of Engineering & Science Research

"CalorieWise: Deep Learning for Food Image Analysis and Calorie Estimation"

Kadari Neeraja [1],
Associate Professor,
JNTUH College of Engineering, Hyderabad.
kadari.neeraja@gmail.com

Bommai Meenakshi [2],
meenakshibomma@gmail.com,

Abstract

In today's fast-paced and work-centric lifestyle, maintaining good health has become a significant concern. People often struggle to find time for themselves, leading to an increasing reliance on quick and convenient meals. As a consequence, keeping track of calorie intake has become a challenging task. To address this issue, our Paper focuses on using deep learning techniques to calculate the approximate calorie count of a food item from an input image. The core of this Paper is a Convolutional Neural Network (CNN) that identifies the food item present in the input image. Once the food is recognized, the system automatically calculates the number of calories associated with that particular item.

This innovative system is particularly beneficial for individuals who aim to follow a strict and healthy diet. By accurately tracking their calorie intake, they can stay on top of their nutritional goals and maintain their fitness effectively. With this solution, people can make informed food choices, even amidst their busy schedules, promoting a healthier lifestyle overall.

Keywords – Life Style-Health-Tightly packed schedule-Healthy meal- Calorie Count of food- Deep learning- Convolution Neural Network (CNN).

INTRODUCTION

In today's fast-paced urban lifestyle, health has become a paramount concern due to limited personal time and the prevalence of on-the-go meal choices, often laden with unhealthy ingredients like fat and sugar. The continuous consumption of such processed foods can lead to various health issues. Monitoring calorie intake has become a challenging task, necessitating a solution like this Paper. The Paper's core objective is to estimate the approximate calorie content of a food item based on an input image. Leveraging deep learning concepts, the system builds neural models for data prediction through three key steps: data collection, feature extraction using classification algorithms, and output prediction based on these extracted features. Fig.1 explains concept of deep learning.

Fig 1: The concept of deep learning

The Paper employs Convolutional Neural Networks (CNN) to accurately identify and predict the scanned input, which is then mapped to a dataset containing calorie per gram information for various food items. The system calculates the number of calories accordingly, enabling users to easily track their calorie intake and maintain a healthy lifestyle. Its primary aim is to provide users with a quick and automated estimation of the calories in the food they consume. To achieve this, the system extracts the food item's name and calorie content alongside image classification using the CNN algorithm. Secure and user-authenticated access ensures the system's usability and reliability.

RELATED WORK

Here are some previously implemented systems related to food image recognition and calorie estimation:

1. In their work "Automatic Calorie Estimation System of Food Images on a Smartphone" presented at the ACMMM Workshop on Multimedia Assisted Dietary Management in 2016[1], K. Okamoto and K. Yanai introduced a novel single-image-based food calorie estimation system. The application runs directly on a Smartphone without the need for external recognition servers. The system performs food region segmentation, categorization, and calorie estimation automatically. As more people focus on healthy eating habits and record their daily diets, this system aims to assist users with a convenient way to estimate calorie content in their meals in paper [2], authors ,Kadari Neeraja, Dr. G. Narsimha. Through experiments and user studies, the effectiveness of the proposed system was confirmed.

2. V. Balaji Kasyap and N. Jayapandian proposed "Food Calorie Estimation using Convolutional Neural Network," where they employed a deep learning algorithm to accurately measure food calories from images. R. Dinesh Kumar, E. Golden Julie, Y. Harold Robinson, S. Vimal4 · Sanghyun Seo [7]. Their Convolutional Neural Network (CNN) model focuses on food object identification to determine calorie values. The primary result parameter is volume error estimation, with calorie error. Similar work was proposed by Kadari Neeraja, Dr. G. Narsimha in their work[3].

estimation being a secondary component. The proposed CNN model achieved a higher level of accuracy compared to other models, gradually reducing volume error estimation by 20%.

3. In "ARDeep Calorie CamV2: Food Calorie Estimation with CNN and AR-Based Actual Size Estimation," presented by R. Tanno, T. Ege, and K. Yanai, a novel approach for food calorie estimation was introduced and also Shota Horiguchi, Sosuke Amano, Makoto Ogawa, Kiyoharu Aizawa, proposed similar kind of work[8]. The system utilizes a combination of Convolutional Neural Network and Augmented Reality (AR) based actual size estimation. By leveraging the Apple ARKit framework, the system measures the actual size of the meal area using three-dimensional vector coordinates in the real world. A kind of research was done by Kadari Neeraja, Dr. G. Narsimha in their work[4]. This approachimproved the accuracy of size

measurement compared to previous direct measurement methods, consequently enhancing the accuracy of calorie estimation as well.

4. B. Preetha Kumari and Dr. G. Wiselin Jiji presented "Food Calories Estimation Using SVM" at the International Conference on Emerging Trends in Application of Computing. **Parisa Pouladzadeh, Shervin Shirmohammadi, and Rana Almaghrabi [9] was done kind of work.** Their system addresses the growing global concern regarding weight management, healthy eating, and obesity prevention by proposing a food calorie nutrition measurement system. The application uses the device's built-in camera to capture an image of the food. A research was done by Kadari Neeraja, Dr. G. Narsimha in their work[5]. A Support Vector Machine (SVM) classifier extracts features from the input food image and classifies it accordingly. Once the food item is identified, the system calculates its calorie content. Kadari Neeraja, Dr. G. Narsimha attempted work[6]. The measurement of calorie intake using this system was found to be more accurate, aiding people in controlling their diet effectively{10-13].

PROBLEM DEFINITION

In our fast-paced work-centric lifestyle, manually tracking calorie intake has become increasingly challenging, leaving little time for health-conscious individuals to monitor their nutritional choices. To address this issue effectively, I have developed a unique system capable of calculating the calorie content of a given food item from an input image.This innovative system relies on Convolutional Neural Network (CNN) technology to classify the input image accurately. By extracting and analyzing the image's features through a series of convolutional and pooling layers, the system identifies the food item with the highest probability. Once the image is classified, the system calculates its calorie content using the per-gram values available in the dataset.By utilizing image recognition and deep learning techniques, this system provides a convenient and precise way for users to stay informed about their calorie intake, promoting healthier dietary habits amidst their busy schedules.

PROPOSED APPROACH

SYSTEM ARCHITECTURE

A system architecture, also known as systems architecture, is a conceptual model that outlines the structure, behavior, and various perspectives of a system. It provides a formal description and representation of the system, enabling reasoning about its structures and behaviors.

The figure fig.2 below illustrates the system architecture. The system is designed to accommodate both administrators and registered users. Administrators can log in using their credentials and, upon successful validation, access the admin home screen. They have the option to upload a dataset containing per gram calorie values and class labels. The dataset can be provided by specifying its path.

Additionally, administrators can view the uploaded dataset to verify its contents. Registered users can log in using their valid credentials and access the user home page. Here, users can input an image of a food item along with its approximate weight through a designated window.

The system then utilizes Convolutional Neural Network (CNN) technology to create a model and compare it with the trained data. The system computes the probability of the input image matching each trained item and predicts its class label. Based on the class label, the system estimates the calories using the per gram values from the calorie dataset and returns the total number of calories to the user. This architecture enables efficient calorie estimation for users based on their input images and weight information.

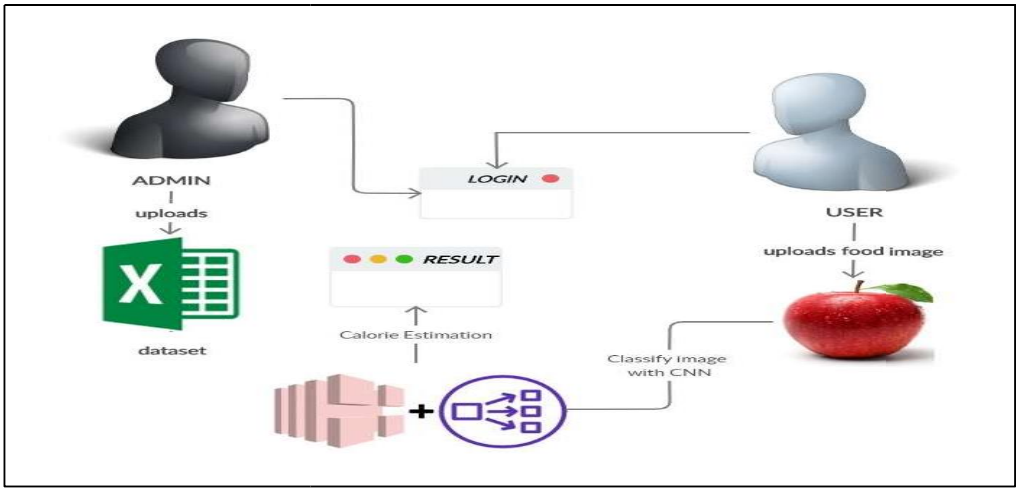

Fig 2: System Architecture

The system architecture is structured into three distinct modules, each serving a specific functionality within the system. These modules are as follows:

1. Admin Module
2. User Module
3. Image Detection Module

3.1.1 **ADMIN MODULE**:

The admin module allows administrators to log in using their credentials. If the entered password or username is incorrect, the login attempt fails. Upon successful login, administrators gain access to the admin home screen, where two options are provided. Admins can upload a dataset containing per gram calorie values by specifying its path. Another option available to the admin is to view the uploaded dataset.

3.1.2 **USER MODULE**:

The user module enables users to sign up and register. Registered users can log in by providing accurate credentials; otherwise, the login attempt fails. Upon successful login, users are directed to the user home page. Within this interface, users can access the uploads tab, where they can submit an image of a food item for calorie estimation. The user provides the path to the input image and inputs the approximate weight of the food item in the image. By clicking the detection button, the user initiates the calorie estimation process.

3.1.3 **IMAGE DETECTION MODULE**:

In the image detection module, the system processes the input image by bringing it into the directory. The image undergoes layers of Convolutional Neural Network (CNN), generating a model for the input image. This model is then compared with the models of each item stored under hierarchical data format. Probabilities are determined, and class labels are sorted. Once the class label is predicted, the system retrieves the per gram calorie value from the calorie dataset uploaded by the admin and performs calorie estimation based on the weight provided by the user.

METHODOLOGY:

The Paper relies on the cutting-edge Convolutional Neural Networks (CNN) algorithm, which has been progressively developed and refined over time, especially in the realm of Computer Vision and Deep Learning. As a powerful deep learning algorithm, CNN takes an image as input and assigns significance, known as weights, to different elements within the image. The algorithm then produces an output, which represents the class or category to which the input image belongs. In essence, CNN excels at extracting essential features from the input image and accurately classifying it based on those features. Its capabilities in feature extraction and image classification have significantly contributed to the advancements in the field of Computer Vision, making it a vital component of this Paper. Below fig.3. Illustrates same.

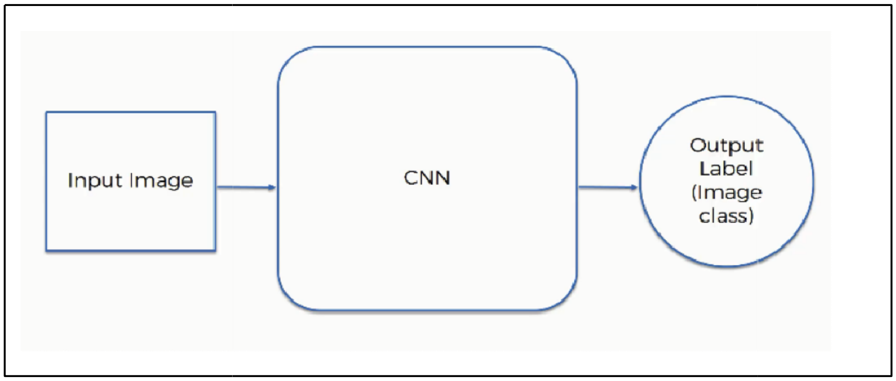

Fig 3: Basic Structure of CNN

A Convolutional Neural Network (CNN) is structured with various node layers, including an Input Layer, one or more Hidden Layers, and an Output Layer.The Input Layer of CNN is responsible for receiving image data as input to the model. This image data is represented by a 3-dimensional matrix that contains pixel values, allowing the CNN to process the pixel data effectively. Subsequently, the data from the Input Layer is passed on to the Hidden Layer, which consists of multiple layers depending on the specific CNN architecture and data size. Typically, the Hidden Layer in a Convolutional Neural Network comprises a series of convolutional layers that perform convolution operations, involving multiplication or other dot products.

4.1 The essential layers in this Hidden Layer are as follows:

4.1.1. Convolutional Layer

4.1.2. Pooling Layer

4.1.3. Flatten Layer

4.1.4 Dense Layer

Each of these layers plays a crucial role in extracting and processing features from the input data, ultimately leading to meaningful outputs or predictions from the CNN. The fig.4. depicts this.

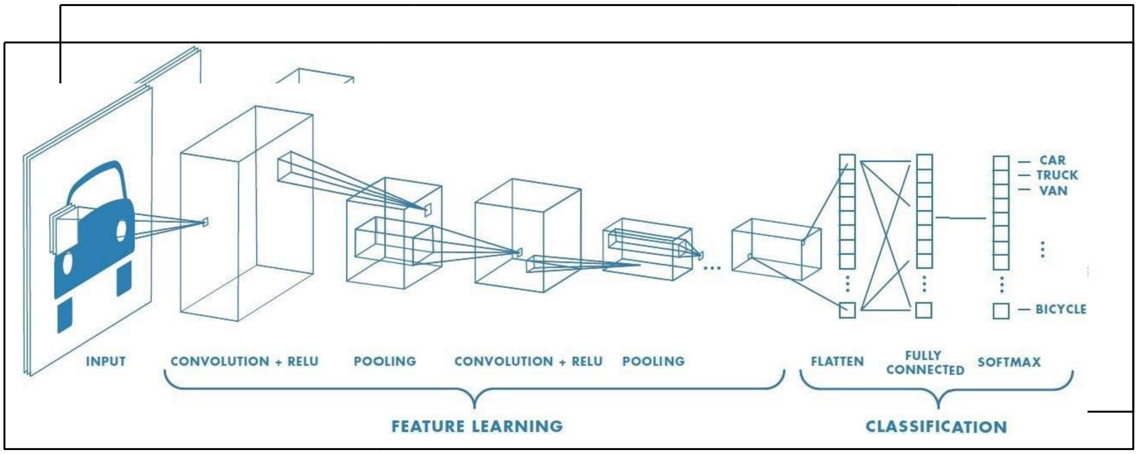

Fig 4: CNN Algorithm Flow

The flow of data begins with the Input Layer, which feeds into the essential building block of CNN known as the convolutional layer. This layer plays a pivotal role in feature extraction from the provided input. It involves three key components: the input data, a filter (also known as a kernel), and a feature map. As the kernel moves across the receptive fields of the input image, it extracts the necessary features through a process called Convolution. This fundamental process is instrumental in enabling CNNs to identify meaningful patterns and information from the input data.

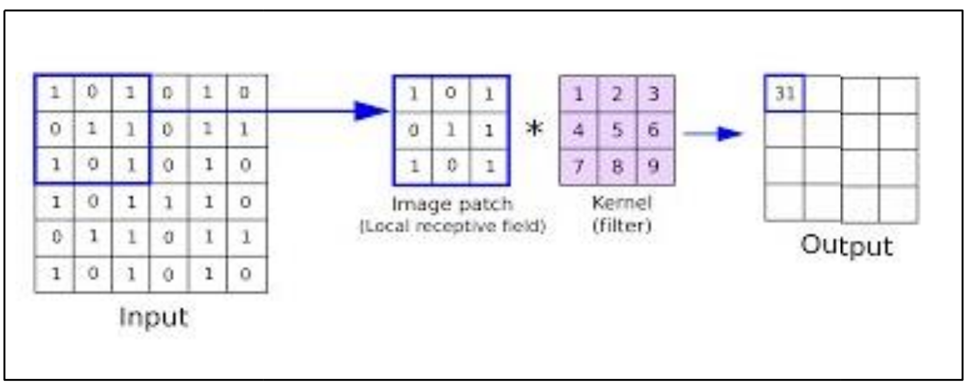

Fig 5: Convolutional Layer Flow

4.1.1 **Convolutional layer:**

During the convolution process, each pixel within the image patch undergoes multiplication with the corresponding filter value, as depicted in Fig 5. The resulting products are then added together. This convolution operation generates a feature map, which effectively highlights the presence of specific features in the input image. The process is iterated for all filters, producing probability matrices containing both positive and negative values.

Following each convolution operation, an activation function is applied to every value in the feature map. In this model, the ReLU (Rectified Linear Unit) activation function is utilized, serving to introduce non-linearity and enhance the model's ability to capture complex patterns and relationships within the data.

The Rectified Linear Unit (ReLU) is an essential non-linear activation function utilized in this layer. Its primary purpose is to introduce non-linearity into the model. During this process, all negative values within the matrices are replaced with zero, while the positive values remain unchanged. This approach effectively prevents the values from summing up to zero, ensuring meaningful transformations in the data. The ReLU activation function can be expressed as follows:

F(x) = 0 if x < 0

F(x) = x if x >= 0

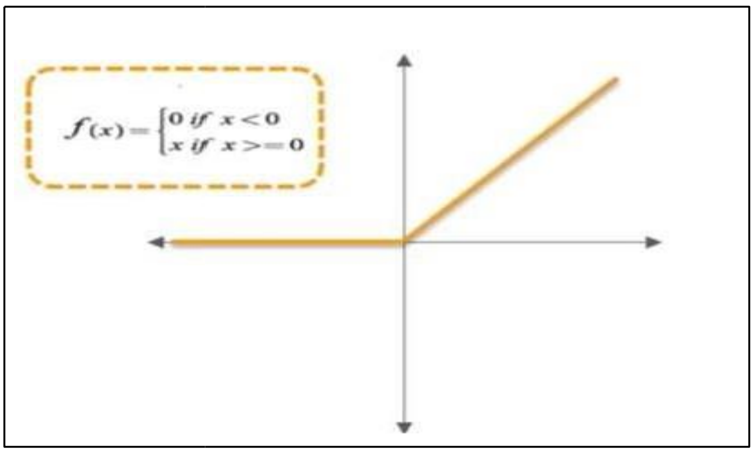

Fig 6: Graph of ReLU function

As a result, each feature map obtained through ReLU activation contains only non-negative values, significantly enhancing the model's ability to capture important patterns and information from the input data as shown in fig.6.

4.1.2. Pooling Layer

The pooling layers in Convolutional Neural Networks (CNNs) are commonly referred to as down-sampling layers. Their primary function is to perform dimensionality reduction, effectively reducing the dimensions of the feature maps and the number of parameters in the input. Various types of pooling layers exist, with one of them being the Max-Pooling layer, which is utilized in this model.

Max-Pooling is specifically designed to extract important features from the feature map. During this pooling operation, the layer selects the maximum value within the region of the feature map covered by the filter. As a result, the output obtained after the max-pooling layer comprises a feature map containing the most prominent and significant features from the previous layer's feature map. This process aids in reducing computational complexity while preserving essential information crucial for subsequent layers' analysis. The fig.7 demonstrates it.

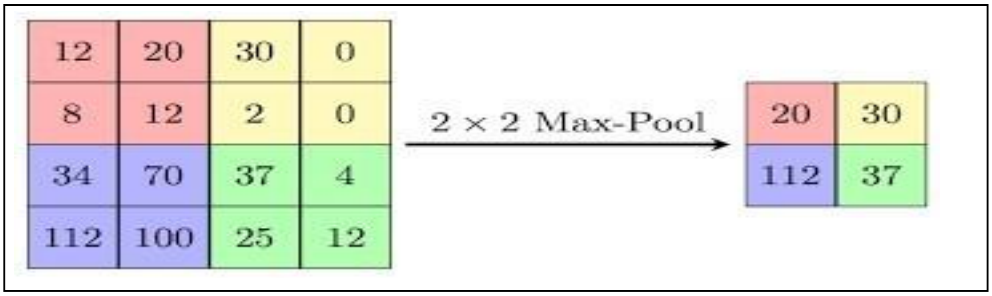

Fig 7: Max-Pooling Layer

4.1.3. **FLATTEN LAYER:**

The process of image classification using a Convolutional Neural Network (CNN) involves several layers. The output from the pooling layer is passed on as input to the flatten layer. Here, the extracted features are transformed into a column matrix, creating a long feature vector. This flattened input is then connected to the final classification model, also known as the fully-connected layer. See fig8 for the same.

Fig 8: Flatten Layer

4.1.4. **DENSE LAYER**

The dense layer, which is another term for the fully-connected layer, plays a crucial role in this setup. It establishes connections between each neuron in one layer to every neuron in the subsequent layer, resembling a traditional multi-layer perceptron neural network (MLP). The flattened feature matrix goes through this fully-connected layer for the purpose of image classification. Each neuron in the dense layer receives inputs from all the neurons in the previous layer, and they collectively contribute to providing outputs to the next layer.fig.9 exhibits this process.

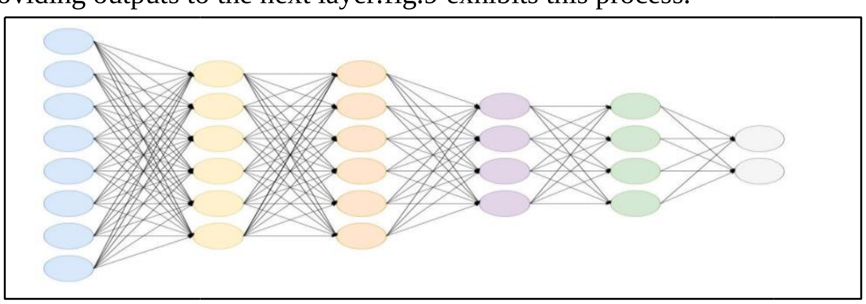

Fig 9: Featuring Fully Connecter Network

The combination of convolutional, ReLU, and pooling layers is responsible for identifying essential features in the input image. On the other hand, the flatten and dense layers are responsible for making accurate classifications based on these extracted features.

The output layer is the final layer of the network, which produces the result of the image classification task. It provides the respective class name for the input image, indicating its predicted category.

V.DATASET

In this Paper, I utilized the 'Food Images (Food-101)' dataset sourced from kaggle.com. The dataset consists of various food images, and I focused on 10 specific food item categories: Beignets, Chicken Wings, Egg Benedict, Fried Rice, French Fries, Macaroons, Mussels, Pancakes, Samosa, and Steak. Each category comprises 1000 images.

To train and evaluate the model effectively, I split the dataset in a 90:10 ratio. This means that 90% of the data was utilized for training the model, while the remaining 10% was reserved for testing the model's performance. This partitioning enables the model to learn from a substantial portion of the data while also providing an independent set of data for evaluation, ensuring a more reliable assessment of its effectiveness.

Fig 10: Dataset

5.1 TRAINING

All models are trained using preprocessed images to classify them into their respective classes. The training process is carried out by the 'train_model' function, which takes a specific model as input and trains it for a specified number of epochs. During training, the model's performance is evaluated on validation data after each epoch, and the function returns the validation accuracy, loss, and accuracy for each epoch.

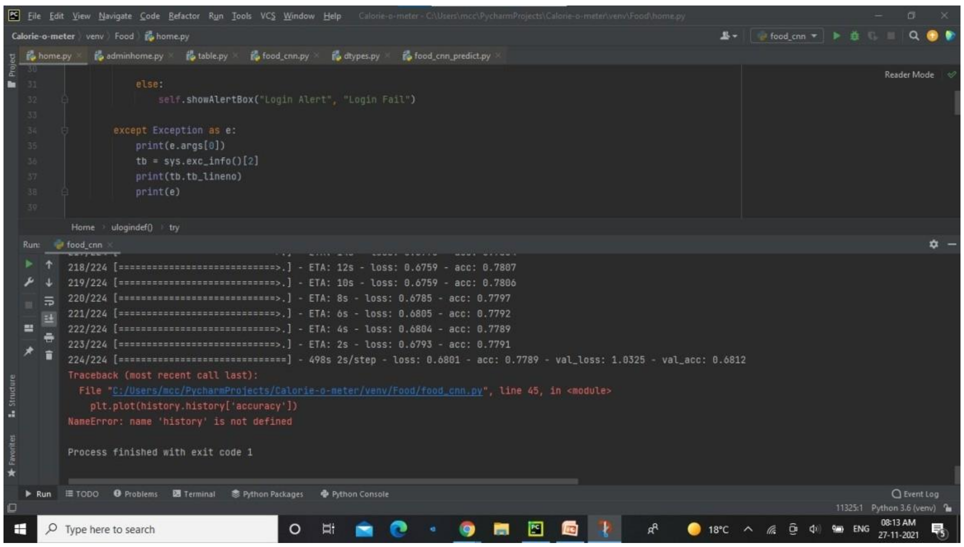

Fig 11: Loss and Accuracy metric

The accuracy metric in the above figure depicts the model's performance across a series of epochs. In this case, the model was trained for 224 epochs. At the end of each epoch, both the loss and accuracy values are recorded. As the training progresses through all the epochs, the validation accuracy is eventually obtained after completing all the iterations.

5.2. CALORIE CALCULATION:

After detecting the name of input image, the next step is to calculate the calorie content of the respective food item. To do this, the user has to provide approximate weight of the food item in grams. And also the system traces the per gram calorie value of the respective food item, that isidentified by CNN model, from the dataset Admin uploads. Then the calorie content is obtained by multiplying the weight and calorie per gram as follows.

ALGORITHM

The algorithm can be explained in various phases and describe each phase step-by-step.

Phase 1: Initialization

1. Import the required libraries (`cv2` and `numpy`).

Phase 2: Define the CNN Class

1. Define the class `CNN` for food category detection.
2. Initialize the model and weights paths (`model_path` and `weights_path`).
3. Create a dictionary `food_categories` to map class indices to food category names.

Phase 3: Load Model

1. Implement the `load_model()` method to load the pre-trained CNN model.
   1. Read the model and weights files from the specified paths (`model_path` and `weights_path`).
   2. Create a model instance using `cv2.dnn.readNetFromTensorflow(model_path, weights_path)`.
   3. Return the loaded model.

Phase 4: Food Category Detection

1. Implement the `detect()` method to predict the food category from the given image.
   1. Read the image from the specified `image_path`.
   2. Preprocess the image by resizing it to (150, 150) and normalizing its pixel values.
   3. Set the preprocessed image as input to the CNN model using `model.setInput(blob)`.
   4. Forward propagate the image through the model and obtain the output probabilities using `model.forward()`.
   5. Flatten the output probabilities to a 1-dimensional array using `np.squeeze(output)`.
   6. Sort the output probabilities in descending order to get the most probable class indices using `np.argsort(output)[::-1]`.

7. Loop through the `food_categories` dictionary to find the detected food category:
   - Check if the probability of the current class is greater than or equal to 0.10.
   - If yes, consider it as the detected category and store it in the `food` variable.
   - Break the loop as soon as a valid category is found.
8. Return the detected food category (`food`).

Phase 5: Main Section

1. In the main section (`if __name__ == "__main__":`):
   1. Create an instance of the `CNN` class.
   2. Specify the path to the image (`image_path`) that you want to classify.
   3. Call the `detect()` method with the given `image_path` to get the detected food category.
   4. Print the detected food category.

RESULTS ANALYSIS

This Paper offers a user-friendly platform that enables the estimation of an approximate calorie count for food items by simply uploading their images. By leveraging a powerful Convolutional Neural Network (CNN) algorithm, the system accurately detects and analyzes the uploaded food images, providing valuable insights into their calorie content. Users can conveniently gain knowledge about the caloric value of the food they consume through this efficient and precise calorie estimation process. The system's output displays the calculated calorie count, empowering users to make informed decisions regarding their dietary choices.

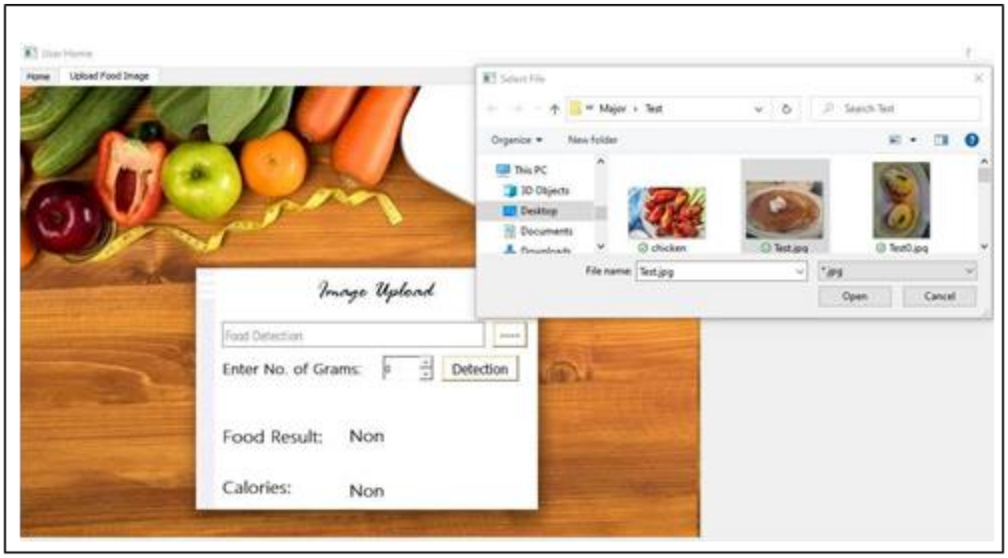

Fig 12: Uploading Image to calculate calorie content

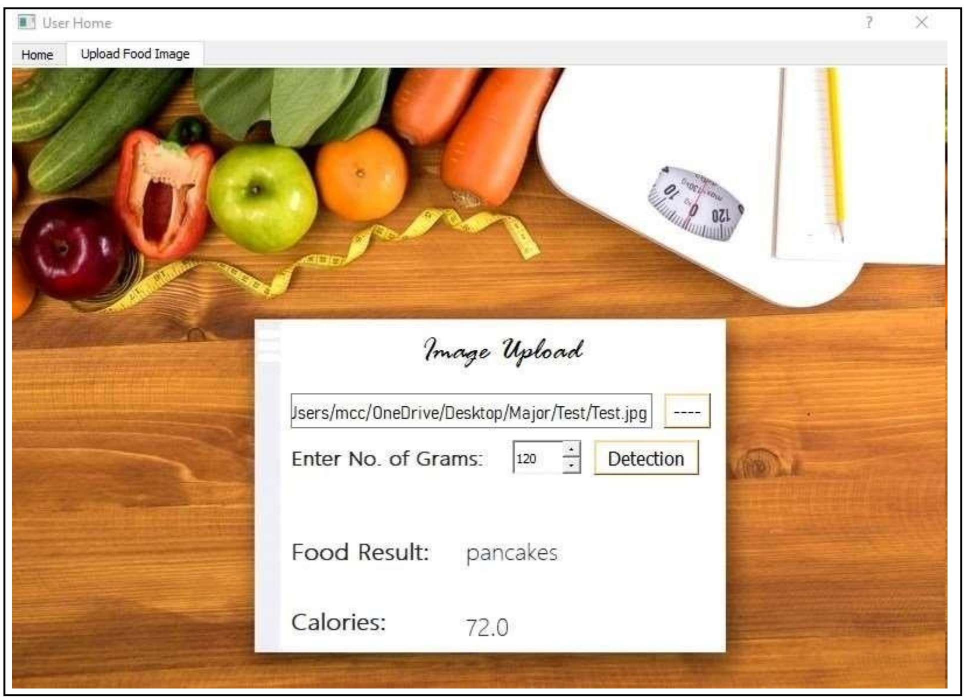

Fig 13: Result of the input image

CONCLUSION

The proposed system aims to estimate the calorie content of food from food images using machine learning techniques. Through experimentation, our calorie estimation approach demonstrates improved accuracy for foods with simple shapes and provides a novel method for estimating calories in foods with irregular shapes. The model utilizes Convolutional Neural Networks (CNNs) to predict food items from input images, enabling accurate calorie calculations based on the identified food names. The implementation of this model, trained on a dataset of food images using CNN, yields results with up to 75% accuracy. This application has the potential to revolutionize how people perceive and manage their food intake, with significant implications for weight-loss and weight-management efforts.

Moreover, there is potential for further enhancement and seamless integration of the Paper into health apps, offering an engineering solution that contributes to users' overall health and well-being. As we continue to refine and develop the system, it holds promise as a valuable tool in the realm of nutrition and dietary awareness, promoting healthier lifestyle choices and empowering individuals to make informed decisions about their food consumption.

REFFERENCES

1. Chang LiuYu Cao,Yan LuoGuanling ChenVinod VokkaraneYunsheng Ma,"DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment", Conference paper,21 May 2016.
2. Kadari Neeraja and G Narsimha, "Multi-Objective Optimal Path Planning for Autonomous Robots with Moving Obstacles Avoidance in Dynamic Environments" International Journal of Advanced Computer Science and Applications (IJACSA), 13(12), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0131226.
3. Kadari Neeraja and G Narsimha, "A MULTI OBJECTIVE HYBRID COLLISION-FREE OPTIMAL PATH FINDER FOR AUTONOMOUS ROBOTS IN KNOWN STATIC ENVIRONMENTS", Scalable Computing: Practice and Experience, ISSN 1895-1767 , Volume 23, Issues 4, pp. 389–402 https://doi.org/10.12694/scpe.v23i4.2049.
4. Kadari Neeraja, N., Narsimha, G. (2023). A Multi Objective Hybrid Collision-Free Near-Optimal Dynamic Path Planner for Autonomous Robots in Dynamic Environments. In: Garg, L., et al. Key Digital Trends Shaping the Future of Information and Management Science. ISMS 2022. Lecture Notes in Networks and Systems, vol 671. Springer, Cham. https://doi.org/10.1007/978-3-031-31153-6_30.
5. Kadari Neeraja, Dr. G. Narsimha. (2022). Hybrid Dynamic Path Planning Approach for Autonomous Robots in partially Known Dynamic Environments. Computer Integrated Manufacturing Systems, 28(12), 361–380. Retrieved from http://cims-journal.com/index.php/CN/article/view/415.
6. Kadari Neeraja, Dr. G. Narsimha, "Multi Objective Hybrid Dynamic Path Planning Approach for Autonomous Robots in Partially Known Dynamic Environments with Moving Obstacle avoidance"- -2022, Vol,54,Issue:3.pg.No: 805-821, https://www.advancedengineeringsciences.com/article/301.html.
7. R. Dinesh Kumar, E. Golden Julie, Y. Harold Robinson, S. Vimal4 · Sanghyun Seo, "Recognition of food type and calorie estimation using neural network", 4 January 2021.
8. Shota Horiguchi, Sosuke Amano, Makoto Ogawa, Kiyoharu Aizawa, "Personalized Classifier for Food Image Recognition," in IEEE Transactions on Multimedia, vol. 20, no. 10, pp. 2836-2848.
9. Parisa Pouladzadeh, Shervin Shirmohammadi, and Rana Almaghrabi, "Measuring calorie and nutrition from food image," IEEE Transactions on Instrumentation and Measurement, pp. 1947–1956, 2014.
10. Horea Muresan, Mihai Oltean, "Fruit Recognition From Images Using Deep Learning" – ACTA Univ. Sapientiae, Informatica, 10, 1 (2018)
11. B. Prretha Kumari, Dr. G. Wiselin Jiji, "Food Calories Estimation Using SVM" – International Conference on Emerging Trends in Application of Computing (ICETAC 2K17).

12. Almaghrabi, G. Villalobos, P. Pouladzadeh, and S. Shirmohammadi, “A novel method for measuring nutrition intake based on food image,” in Proc. IEEE Int. Instrum. Meas. Technol. Conf., Graz, Austria, May 2012, pp. 366–370.
13. Rahul Chauhan, Kamal Kumar Ghanshala, R. C. Joshi, “Convolutional Neural Network (CNN) for Image Detection and Recognition” – IEEE, 2018 First International Conference Publication.