ANALYSIS OF FOOD RECOGNITION AND CALORIE ESTIMATION USING AI

A project reports

Submitted in the partial fulfillment of the

requirements for the award of the degree of

Bachelor of Technology

in

Department of **ELECTRONICS AND COMMUNICATION**

by

M.NVVSLD SANTHOSHI	170040519
R. VAMSI	170040735
U. TEJASRI	170040889

Under the supervision of

B. Kali Vara Prasad

(Ass.Prof)

Dept of ECE KLEF

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation

(Deemed to be University estd., u/s 3 of UGC Act 1956)

Greenfields, Vaddeswaram, Guntur (Dist.), Andhra Pradesh - 522502

March, 2021

Declaration

The Project Report entitled "ANALYSIS OF FOOD RECOGNITION AND CALORIE ESTIMATION USING AI" is a record of Bonafide work of M. SANTHOSHI (170040519), R. VAMSI (170040735), U. TEJASRI (170040889) submitted in partial fulfillment for the award of B.Tech in Electronics and Communication Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

M.SANTHOSHI (170040519)

R. VAMSI (170040735)

U. TEJASRI (170040889)

Certificate

The Project Report entitled "ANALYSIS OF FOOD RECOGNITION AND CALORIE ESTIMATION USING AI" is a record of bonafide work of M. SANTHOSHI (170040519), R. VAMSI (170040735), U. TEJASRI (170040889) submitted in partial fulfillment for the award of B. TECH in Electronics and Communication Engineering to the K L University.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of the Supervisor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGEMENT

Apart from the efforts of ours, the success of any work depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project-based lab report.

We express our gratitude to our Project Guide **Mr. KALI VARA PRASAD (Asst.Prof), Dept of Electronics & Communications Engineering.** We can't thank him enough for tremendous support and help. We feel motivated and encouraged every time we attend his meeting. Without his encouragement and guidance this Project would not have materialized.

We are thankful to our Head of the Department **Dr. SUMAN MALOJI, Professor, Dept of Electronics & Communications Engineering** who modeled us both technically and morally for achieving greater success in life.

We express sincere gratitude to our Course Coordinator **Dr. BHAVANA** for his leadership and constant motivation provided in successful completion of our academic semester**.**

Finally, we owe a lot to the teaching and non-teaching staff of the Dept. of ECE for their direct or indirect support in doing our Lab based project work.

ABSTRACT

Food is essential for human survival and has been the subject of several medical conventions. New dietary assessment and nutrition research resources are now available, allowing people to better consider their everyday eating habits, investigate nutrition trends, and sustain a balanced diet. Many dieticians and healthcare professionals are concerned with consuming the correct amount and type of food . In addition to physical activity and exercises, maintaining a healthy diet is necessary to avoid obesity and other health-related issues, such as diabetes, stroke, and many cardiovascular diseases. Recent advancements in machine learning applications and technologies have made it possible to develop automatic or semi-automatic dietary assessment , which more convenient approach to monitor daily food control eating habits. These solutions aim to address the issues found in the traditional dietary monitoring systems that suffer from imprecision, underreporting, time consumption, and low adherence. In this project, the recent vision-based approaches and techniques have been widely explored to outline the current approaches and methodologies used for automatic dietary assessment, their performances, feasibility, and unaddressed challenges and issues.

Convolutional Neural Network (CNN) was the integrated recognize food on an image. Graph Cut image segmentation was used to analyze to determine the regions of the food in the image. Volume estimation based its measurement on the area of the segmented food image. Training part of the prototype system. The main purpose of the proposed method is to improve the accuracy of the pre-training model. The paper designs a prototype system based on the client server model. The client sends an image detection request and processes it on the server side. The prototype system is designed with three main software components, including a pre-trained CNN model training module for classification purposes. We experimented with a variety of food categories, each containing thousands of images, and through machine learning training to achieve higher classification accuracy.

The use deep convolutional neural networks to classify 10000 high-resolution food images for system training. Our results show that the accuracy of our method for food recognition of single food portions is 99%. The analysis and implementation of the proposed system are also described the project.

TABLE OF CONTENTS

CONTENT	PAGE NO

CHAPTER 1

INTRODUCTION

Obesity and overweight are denned as the result of energy calories intake and expenditure [1]. This has been related to the risks of developing chronic heart diseases, diabetes, and other vascular syndromes. Obesity was the leading cause of death in 2012, with more than1.9 billion overweight adults, and 650 million of those were obese [2]. Nutritionists attempt to address these issues traditionally by analyzing and monitoring the daily eating habits of their patients or alternatively by examining the images of consumed food [3]. However, the results are affected by the lack of correct logging of food intake by the patients or by the imprecision in estimating the portion size by simple examination of the food images.

Because people are very keen on measuring weight, healthy diets, and staying away from obesity, there is an increasing demand for food calorie measurement. Adult obesity is increasing at an alarming rate. The main source of obesity is the difference between dietary intake and the energy people get from the diet. High-calorie intake may be injurious and lead to various diseases. Breast, colon and prostate cancers are caused by high calorie intake. High calorie intake is the second leading cause of cancer. Dietitians have determined that the standard intake of a number of calories is required balance to the human body. As reported health organization, more than 110th of the adult population in the world is obese. Obesity is a medical condition in which excess body fat has accumulated to the extent that it may have a negative effect on health [1]. The food a person takes daily is higher utilized then we can say that the respective person is becoming obese. Obesity and being overweight are interconnected to many dangerous and chronic diseases. In 2013, the American Medical Association officially declared obesity as the disease that has serious consequences on patients health and therefore requires medical treatment [2]. Therefore, daily intake measurements are important for losing weight and maintaining a healthy diet for normal people. Only a timely measurement of daily food consumption can make obese people lose weight in a healthier way, and can also make healthy people better healthy. The traditional method is mainly based on the analysis of the user's record, and the clinical display has certain effects, but these methods often cause the patient's uneasiness to be forgotten by the user or the broadcast that the user does not want to use these programs [3].

The proposed scheme would allow people to prepare well for their daily calorie consumption, not only obese people but also healthier people. In the following paragraphs, we can make a contribution to this thesis.

• We propose a transfer learning based novel system that automatically performs the exact classification of the food image and estimates the food attributes.

• We present the dataset for evaluating current system and other deep learning-based recognition systems that will be developed in the future.

• There is no data set that contains subcontinental dishes available to the public, we created a new set of data that includes both subcontinental and other common cuisines.

1.1. PRE-TRAINED MODEL SELECTION

Here we divide our proposed methodology intro three separate parts. The first part has to deal with the transfer learning-based CNN models, the second part has to do with the text recovery from different sources while the third part has to deal with the text data training.

1.2. PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK MODEL

A pre-trained network model is used in machine learning that the system gets stuck in local solution while in its training age. These models can carry out machine training to respond immediately to different data. A CNN model that we used in our suggested process of transferring learning -based food recognition and extraction attributes uses a variety of food items from our prepared dataset to get different characteristics from an object [5].

1.3. DATASET PREPARING AND PER-PROCESSING PHASE

To obtain characteristics from various foods we assign for our research, we categorize each image into its corresponding class. To this end, with the help of different attributes, we distinguish each and every class. For our study, the size of the text data we receive from the internet is nearly 1.8 GB. We used two completely different frameworks to gather data. Common Crawl [6] is the first and Scrapy [7] is the second. We collected about 100 MB of data using Scrapy while using Common Crawl we collected 1.7 GB of data. The dataset we created includes hundreds and thousands of pictures of various foods. For our research study, some images are relevant and some are not. Filtering the data set is remarkable in the preparation of a model. We use the Data Augmentation concept to improve the efficiency of training data. We perform image transformation in data augmentation [8]. To train transformation parameters, we implement Spatial Transform Network [9]. Once the training is complete, these parameters are applied to the image of the food and the image is transformed.

1.4. FOOD IMAGE CLASSIFICATION

A basic automatic dietary assessment system is required to identify and recognize the food contained in a meal. The image classification, a machine learning technique, is used to identify a set of unknown objects that belong to a subset (class), which has been learned by the classifier in the training phase. In this step, food images are used as input data to train the classifier. An ideal classier must be able to recognize any food type that has been included in the learning process. Practically, multiple variations exist in digital images, including rotation, distortion, color distribution, lighting conditions, and so forth, which may affect the overall accuracy. The training process itself is a tedious task that consumes a reasonable quantity of time to reach its intended accuracy goals. The classier accuracy is affected mainly by the quantity and quality of images used in the training process the proper selection of visual features. The extraction of image features used in the learning process splits a typical image classier implementation into two strategies: traditional classier with handcrafted features and deep learning

CHAPTER 2

2. LITERATURE SURVEY

Author	Title	work	Applications/ limitations
T. Ege and K. Yanai	Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions," in *Proc. Thematic Workshops*, 2017, pp. 367_375.	Image based calories estimation on cooked food	Calories estimation only for cooked food
T. Miyazaki, G. C. de Silva, and K. Aizawa,	Image-based calorie content estimation for dietary assessment," in *Proc. IEEE Int. Symp. Multime- dia (ISM)*, Dec. 2011, pp. 363_368.	Calories estimation for dietary assessment by using mobile app	dietary assessment by using mobile app
K. Yanai and Y. Kawano	Food image recognition using deep convolutional network with pre-training and _ne-tuning," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun./Jul. 2015.	Image recognition is done by using deep convolutional network	deep convolutional network application
Q. Bojia	*Food Recognition and Nutrition Analysis Using Deep CNNs*. Montreal, QC, Canada: McGill Univ., 2019.	Analysis of food by deep cnn	Nutrition analysis and estimation
S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar	"Food recognition using statistics of pairwise local features," Computer Vision and Pattern Recognition, IEEE Conference on, pp. 2249 – 2256, 2010.	Analysis and food recognition is done on statistics of pairwise	Computer Vision and Pattern Recognition application
M. Weber, M. Welling, and P. Perona,	Unsupervised learning of models for recognition," Computer Vision, European Conference on, pp. 18 – 32, Jan 2000.	Food recognition on unsupervised learning	unsupervised learning Computer Vision application

C. Martin, S. Kaya, and B. Gunturk	Quantification of food intake using food image analysis," Engineering in Medicine and Biology Society.Annual International Conference of the IEEE, pp. 6869 – 6872, 2009.	food image analysis on Quantification of food intake in medicine	Quantification of food intake in medicine
W. Wu and J. Yang,	Fast food recognition from videos of eating for calorie estimation," Multimedia and Expo, IEEE international Conference on, pp. 1210 – 1213, Jan 2009.	Recognition of fast food calorie estimation	Fast food estimation of clories
M. Anthimopoulos, L. Gianola, L. Scarnato , P.Diem , and S. Mougiakkou,	A food recognition system for diabetic patients based on an optimized bag of features model," IEEE Journal of Biomedical and Health Informatics, 2014.	Food recognition for diabetic patients optimized model	diabetic patients optimized model
L. Bossard, M. Guillaumin, and L. Van Gool,	``Food-101_mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 446_461	``Food-101_mining discriminative components with random forests	Dataset analysis with random forests
Y. Kawano and K. Yanai,	``Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. ECCV Workshop Transferring Adapting Source Knowl. Comput. Vis. (TASK-CV)*,2014, pp. 3_17.	Automatic expansion of a food image dataset existing categories with domain	image dataset existing categories with domain
K. Yanai and Y. Kawano,	``Food image recognition using deep convolutional network with pre-training and _ne-tuning," in *Proc. IEEE Int. Conf.Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1_6.	Food image recognition using deep convolutional network by pre trained	deep convolutional network by pre trained food dataset
Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi,	**"Understanding of a convolutional neural network",** *Department of Computer Engineering, Istanbul Kemerburgaz University, Istanbul, Turkey, August 2017.*	Cnn model	Concept of cnn

CHAPTER 3

3. CNN

(CNN) extricate the highlights of a picture and diminish the size without loss of its qualities. In any case, by utilizing (CNN) we can diminish the quantity of neurons with the assistance of the conv and pooling layers, where we can lessen the quantity of calculations in a completely associated network[3]. In profound learning, a convolutional neural organization is a class of profound neural organizations, most regularly applied to examining visual symbolism. They are otherwise called move invariant or space invariant fake neural networks, based on their shared-loads engineering and interpretation invariance characteristics

(CNN) extract the features and reduce the size without loss of its characteristics. For example, if we have an image with a size of 35x35x1(where 1 represents only 1 channel in RGB). The total number of Neurons for the fully connected layer will be 35*35= 1125. if the image size is 100x100 we need 10000 neurons as input for the fully connected layer, where it was very difficult to compute that many layers [2]. But by using Convolution neural networks (CNN) we can reduce the number of neurons with the help of the conv and pooling layers, where we can reduce the number of computations in a fully connected network

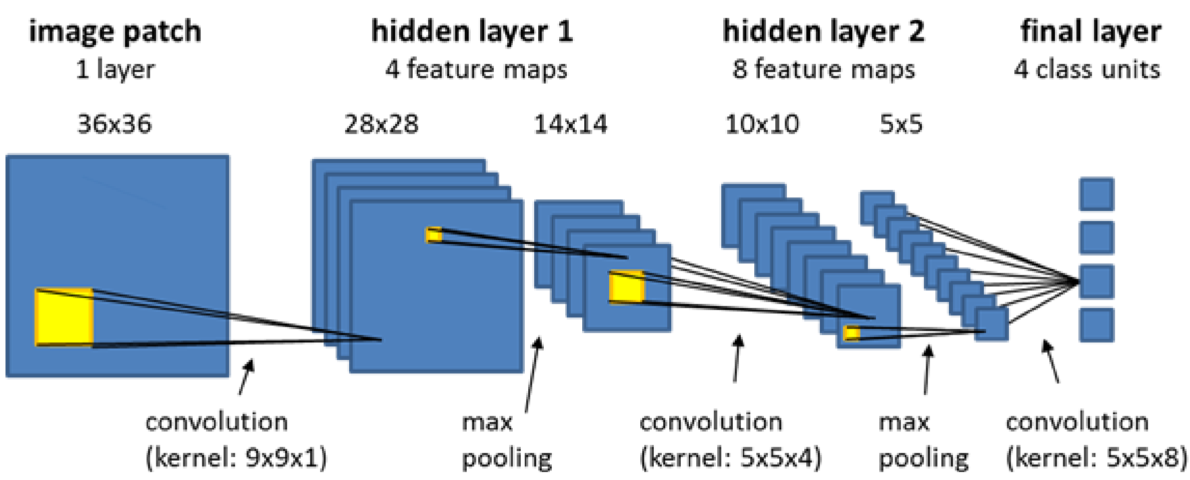
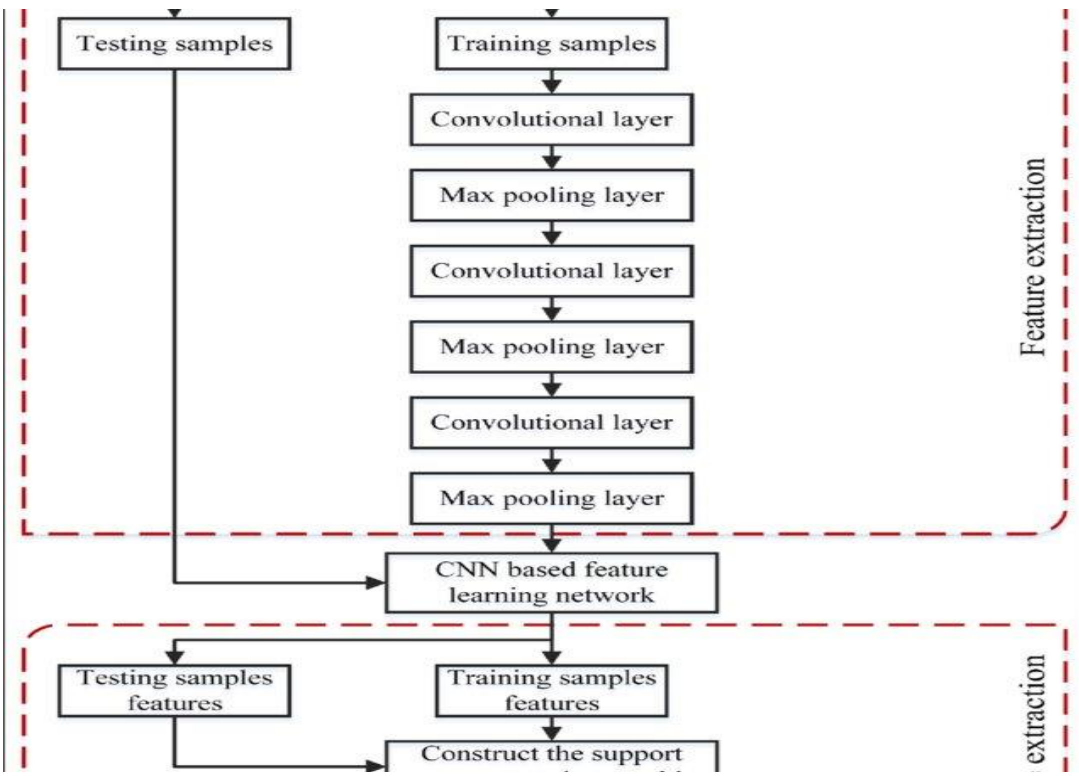


FIG:1: layering of convolution networking

- Convolutional neural network is a class of deep learning methods which has become dominant in various computer vision tasks and is attracting interest across a variety of domains, including radiology.
- Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm.
- Familiarity with the concepts and advantages, as well as limitations, of convolutional neural network is essential to leverage its potential to improve radiologist performance and, eventually, patient care**.**

CNN architecture is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain. The neurons within a **CNN** are split into a three-dimensional structure, with each set of neurons analyzing a small region or feature of the image



3.1. Layers

In Convolution Neural Networks (CNN) we have 5 Layers. they are the input layer, convolution layer, pooling layer, Fully connected layer, and output layer.

3.1.1. Input layer

The input layer contains image data. the images are given in a three-dimensional matrix, we need to reshape the images as a single dimension. suppose if we have an image size of 30x30x1=900 Then need to reshape as 900x1. If you have "n" training examples input will be (900, n).

3.1.2. Convo Layer

The convo layer is called a feature extractor because the image features have extracted within the layer. The edge detection has to be done in the convo layer [4]. The convolution operation has been taken between image and kernel. the kernel is either a horizontal edge filter, vertical edge filter, or diagonal edge filter. Based on the filter the edges will detect. first, A part of the image is convoluted with the kernel, based on kernel size and same as sliding window or dot product total image will convolve as shown in Figure 2.

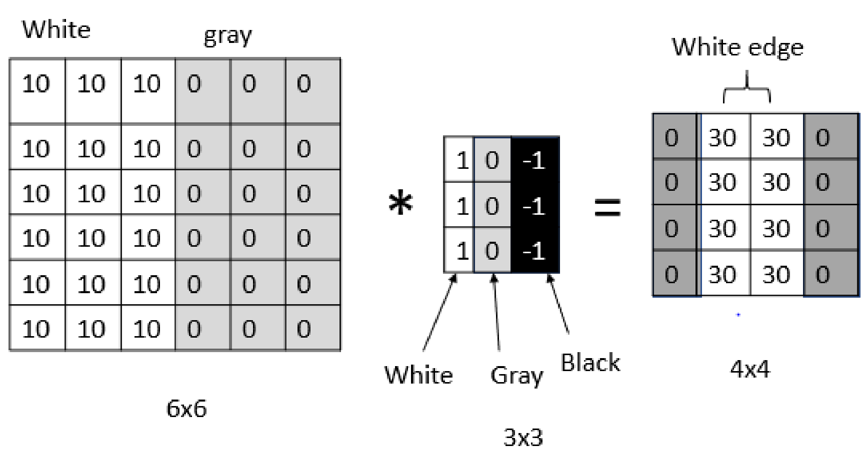


Figure 2: Convo operation with horizontal edge detection

*(N x N) * (F x F) = (N-F+1) x(N-F+1)* 8

Formula for matrix size after image gone through convolution layer

Here the *N x N* is an Image size and *F x F* is the Filter size a kernel size. By convoluting both results an image size (N-F+1) x (N-F+1). Like the above example, the input size is *6 * 6*, the kernel size is *3* 3* the output size is *4 * 4*.

There are also different types of filters like the Sobel filter, Scharr filter, etc., based on the requirement we use them.

3.1.3. **Pooling layer**

Pooling layer between two convo layers, used to reduce the size which is the output of the convo layer. Without the pooling layer, the input for a fully connected network is very computationally Costly, we need to reduce .The max-pooling is the only way to reduce the size of images. In max-pooling, we use stride

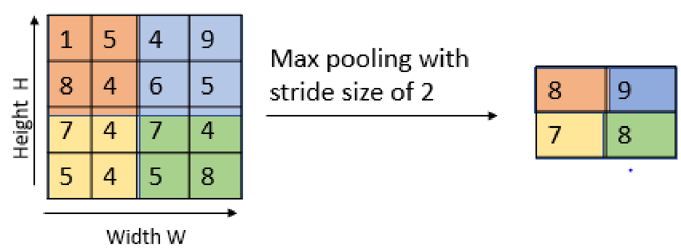


Fig 3. Max pooled with a filter size of 2 x 2 and with a stride of 2

W2=(W1-F)/S+1

H2=(H1-F)/S+1

D2=D1

Where *W1, H1, D1* is width, Height, the dimension of the image, and *W2, H2, D2* are the width, Height, dimension of the max pooled image.

3.1.4. **Fully Connected Layer**

The fully connected layer is used to classify the images between several images by training.it contains neurons, weights, and bias. in a fully connected layer, we have few layers, all the neurons are interconnected as shown in figure 4.

The Last layer Contains the Logistic/SoftMax Layer, basically logistic is used for binary classification and SoftMax is for multi-classification.

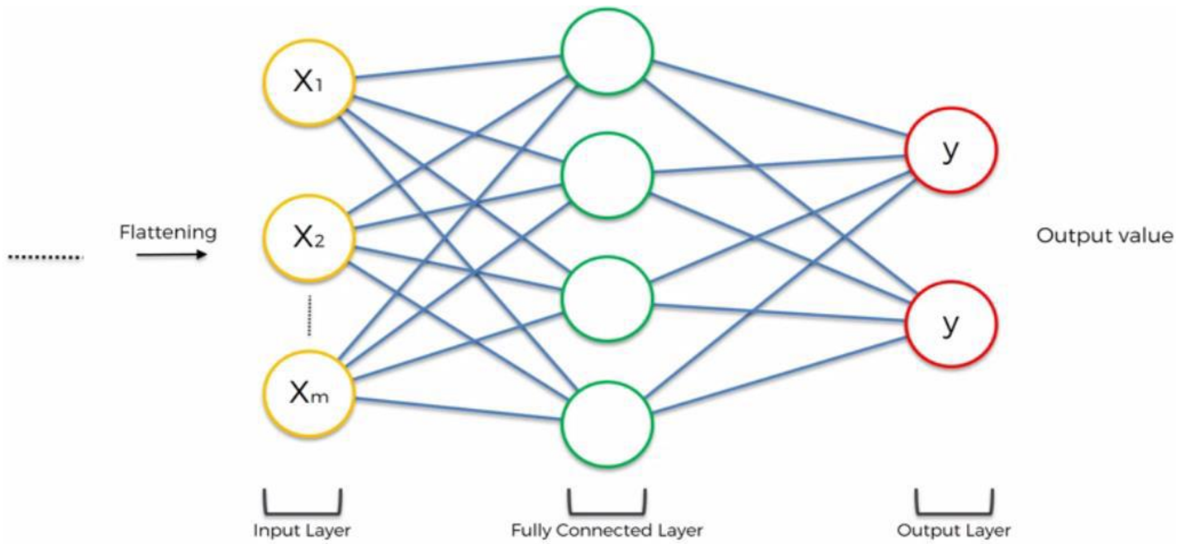


Figure 4: Example of fully connected layer

3.1.5. **Output Layer**

The Output layer presents the obtained results in binary form. If the image is FOOD image the model shows the result as 0 or else it shows normal.

3.2. Image Processing

An image Processing module contains direct and non – linear image sifting and image changes. Image Processing is most usually named as 'computerized Image Processing' and the space wherein it is habitually utilized is 'PC Vision’. Both Image Processing calculations and Computer Vision (CV) calculations accept a picture as info, notwithstanding, in picture preparing the yield is additionally a picture, while in PC vision the yield can be some data about the image.

3.3. Image Representation

The pixels are packed in matrices as height x width x number of channels (for colored image 3 channels RGB as in Figure 3). Based on the red, green, and Green channels intensity the pixel values may vary. for the grayscale images, there will be only one channel.

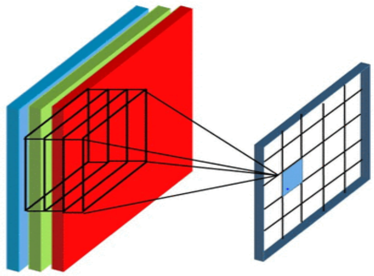
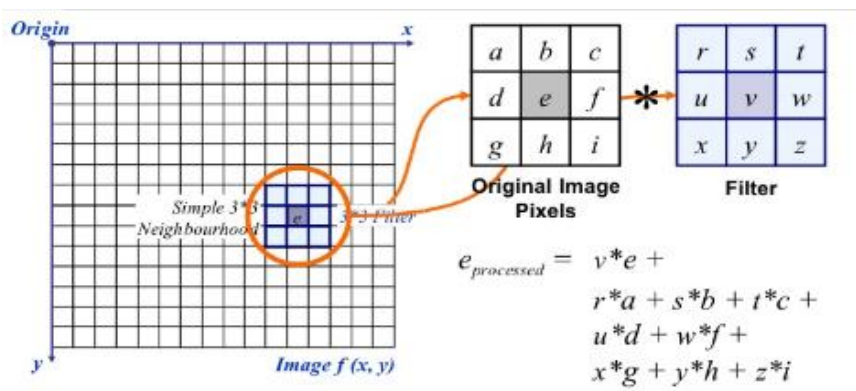


FIG:5: grayscale image

3.4. Image Filtering

In image filtering, a pixel esteem is refreshed utilizing its neighbouring qualities. Yet, how are these qualities refreshed in any case .Every technique has its own employments. For instance, averaging the pixel esteems in an area is utilized for picture.



Gaussian filtering is additionally utilized for picture obscuring that gives various loads to the neighboring pixels dependent on their separation from the pixel viable.

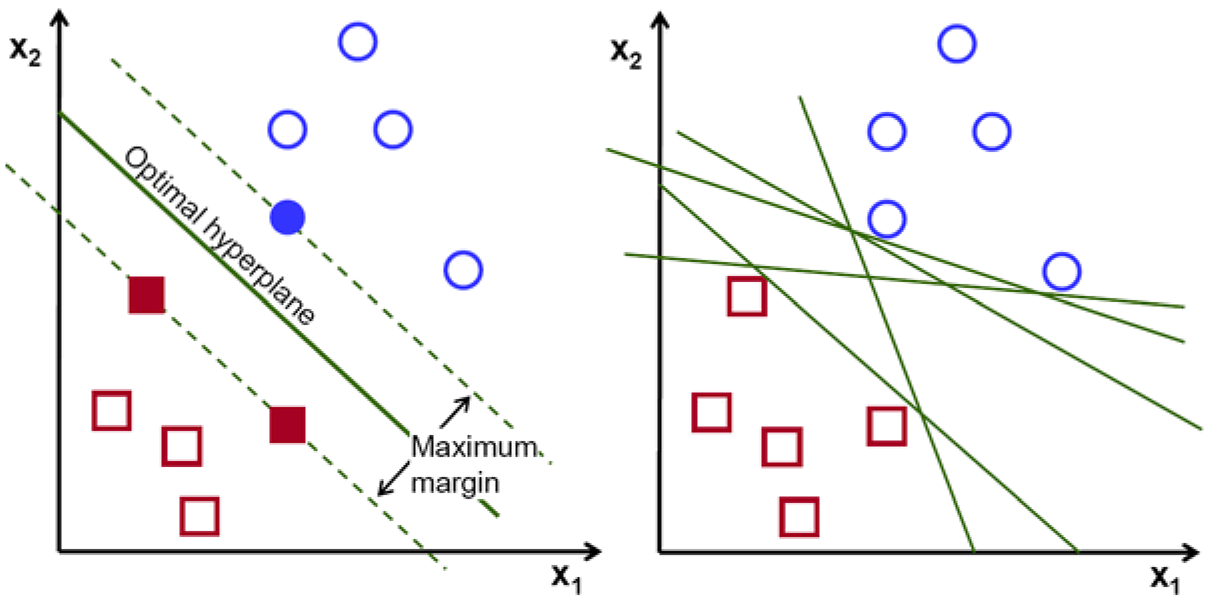
For picture separating, we use parts. Portions are frameworks of quantities of various shapes like 3 x 3, 5 x 5, and so forth A portion is utilized to compute the spot item with a piece of the picture. While figuring the new estimation of a pixel [8], the part community is covered with the pixel. The neighboring pixel esteems are duplicated with the comparing esteems in the piece. The determined worth is relegated to the pixel agreeing with the focal point of the bit.

CHAPTER 4

4. SVM

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

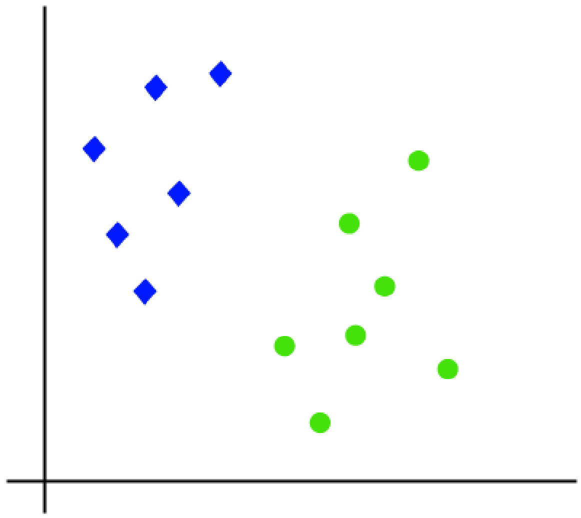
The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.



To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

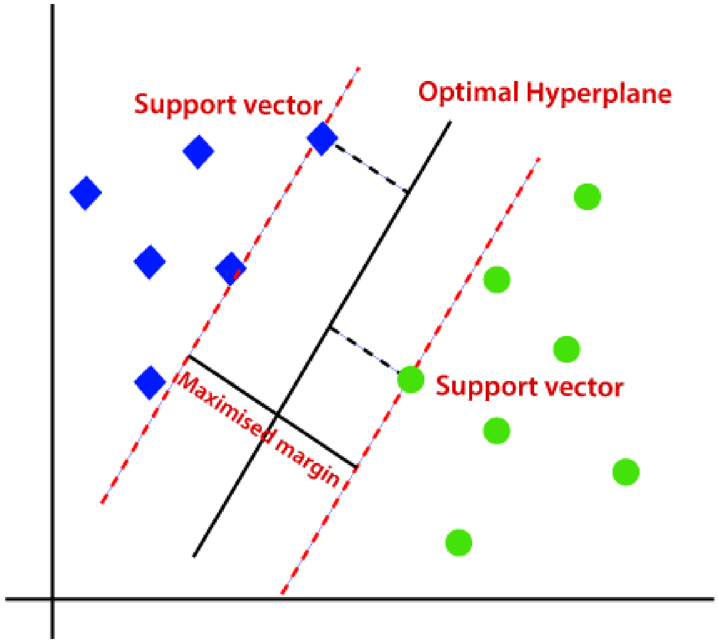
4.1. LINEAR SVM

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



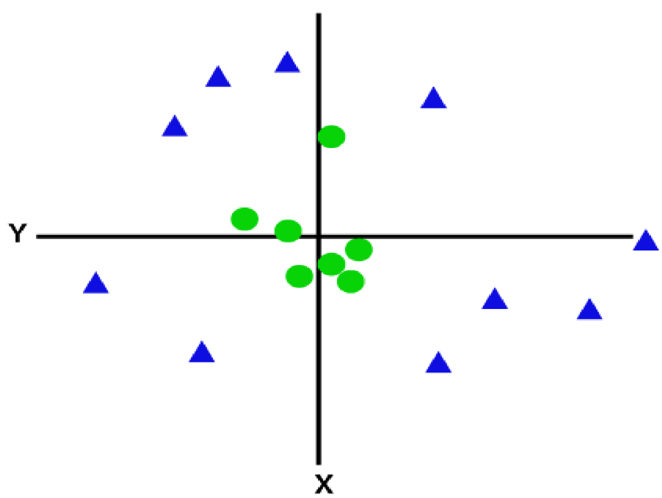
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



4.2. NON-LINEAR SVM

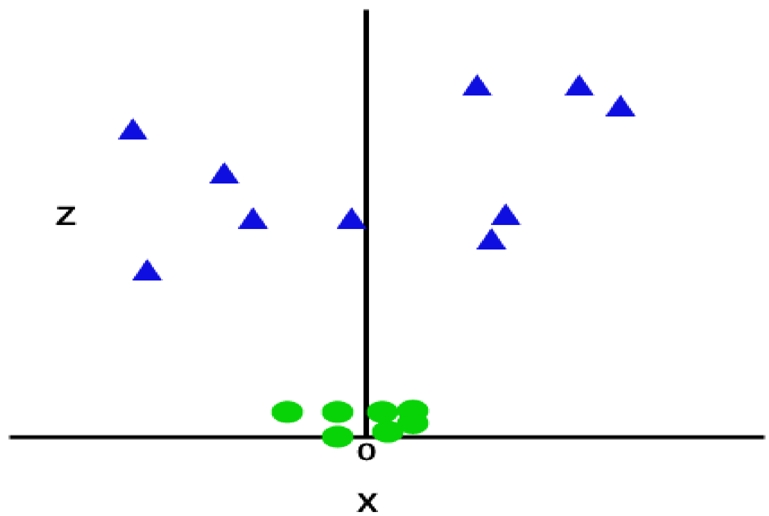
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



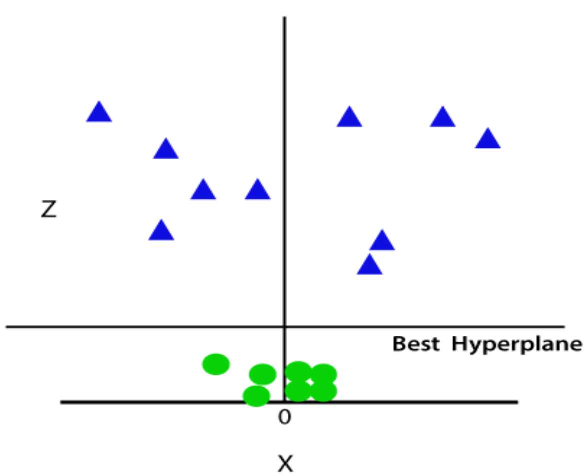
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$$z=x^2 +y^2$$

By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



4.3. Hyperplanes and Support Vectors

Hyperplanes are decision boundaries that aid in data classification. Different groups may be allocated to data points on either side of the hyperplane. The hyperplane's dimension is also defined by the number of functions.

If there are only two input features, the hyperplane is just a line. The hyperplane becomes a two-dimensional plane when the number of input features reaches three. When the number of features reaches three, it becomes impossible to picture.

Updates to the Cost Function and Gradient
The goal of the SVM algorithm is to maximise the distance between the data points and the hyperplane. Hinge loss is a loss feature that aids in margin maximisation.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

Loss of Hinge Feature (function on left can be represented as a function on the right)

If the expected and actual values have the same symbol, the cost is zero. If they aren't, the loss value is calculated. A regularisation parameter is also applied to the cost function. The regularisation parameter's target is to strike a balance between margin maximisation and loss. After the regularisation is applied,parameter, The cost functions appear as shown below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Loss function for SVM

To find the gradients, we take partial derivatives with respect to the weights now that we have the loss function. We can change our weights by using the gradients.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \quad = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \left(1 - y_i \langle x_i, w \rangle\right)_+ \quad = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Gradients

We just need to change the gradient from the regularisation parameter when there is no misclassification, i.e. our model correctly predicts the class of our data point.

$$w = w - \alpha \cdot (2\lambda w)$$

Gradient Update — No misclassification

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularisation parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Gradient Update — Misclassification

In machine learning **support-vector machines** (**SVMs**, also **support-vector networks**) are supervised learning models with associated learning algorithms that analyse data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir, being based on statistical learning frameworks or VC theory proposed

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabeled, supervised learning is difficult, so an unsupervised learning solution is necessary, in which the data is grouped automatically into groups and new data is mapped to these groups.

Have Siegelman and Vladimir Vapnik developed the support-vector clustering algorithm, which uses the statistics of support vectors developed in the support vector machines algorithm to categorise unlabeled data. It is one of the most commonly used algorithm.

APPLICATIONS

SVMs have various real-world problems:

- SVMs may also be used to perform image classification. After just three or four rounds of validity reviews, SVMs reach considerably higher search precision than conventional query refining systems, according to experimental findings. This is also valid for image segmentation schemes, particularly those that use a tweaked version of SVM that employs Vapni's privileged methodology.
- In both the regular inductive and transudative environments, SVMs will greatly reduce the need for labelled training instances, rendering them useful in text and hypertext categorization. Help vector machines are used in several superficial semantic parsing methods.
- The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models. Support-vector machine weights have also been used to interpret SVM models in the past. Postdoc interpretation of support-vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

CHAPTER 5

5. MATERIALS AND METHODS

5.1. Dataset

The number of food pictures we use in this work is 6512. We use ordinally food photos which were uploaded to Food Log by users. While using a dataset that has perfectly correct calorie data is the best, making such a dataset is extremely tedious. Therefore, all the food pictures were examined by experts in nutrition, and their calorie content values were determined. If the food in the picture is for 3 people, the calorie value of the data is 1/3 of the real value. This dataset is our ground truth dataset. We use 5512 pictures from the 6512 pictures as a dictionary dataset and the other 1000 pictures as a test dataset. Sample pictures from the dataset.

The Dataset used in this model contains two folders FOOD and Normal. The FOOD images obtained from a GitHub respiratory contains a total of 1,140 images which are a collection of some food and fruits items. For FOOD images there are two views FRUITS and COOKED FOOD views, The Normal images were obtained from the Pneumonia dataset found on the Kaggle website. There are more than 500 images but we need the images to count equal to COVID images.

5.2. Design of Experiments

A few sorted examinations were performed to assess the effectiveness of the ConvNet on the considered picture information base and to contrast ConvNet and different models utilizing the essential factual attributes of the pictures, which can give 10 powerful data to order. Examinations were partitioned into three classifications: ConvNet tests, factual estimation trials, and move learning tests.

5.3. ConvNets Experiments

ConvNet tests were performed on two subcategories: FOOD/Normal. They incorporated the utilization of four diverse organization models with fluctuating quantities of convolutional and completely associated layers, and fundamental picture pre-handling procedures to test the outcomes utilizing different structures and pre-preparing strategies.

The principal structure (ConvNet#1) comprised of two convolutional layers with 64 and 16 channels, individually, with two completely associated (thick) layers with 128 and 8 neurons. It was the lightest engineering considered in this investigation. The second and third ConvNet structures (ConvNet#2 and ConvNet#3) included three convolutional layers with 256, 128, 64 and 128, 64, 32 channels, individually, and two completely associated layers were executed with 128 and 8 neurons. ConvNet#4, which was the most profound engineering in this investigation, comprised of four convolutional layers (256, 128, 128, and 64 channels) and three completely associated layers (128, 64, and 8 neurons). The channel sizes were considered as 3×3 for all structures, and 0.2 dropout was utilized for each layer. Pooling was applied as greatest pooling, and 2×2 pooling was considered for each layer aside from the last convolutional layer of each structure. The pooling was applied as 1×1 in the last convolutional layer of each structure, to not limit the highlights removed by convolutional layers. Table 1 and Table 2 show the design properties of four thought about ConvNets.

Table 1. Architectural Properties of Four Considered ConvNets.

Architecture Name	ConvNet Layer No.	Filters	Filter Size	Pooling and Size	Dropout	Activation
ConvNet#1	ConvNet Layer 1	64	3×3	Max-pooling 2×2	0.2	ReLU
	ConvNet Layer 2	16		Max-pooling 1×1		
ConvNet#2	ConvNet Layer 1	128	3×3	Max-pooling 2×2	0.2	ReLU
	ConvNet Layer 2	64				
	ConvNet Layer 3	32		Max-pooling 1×1		
ConvNet#3	ConvNet Layer 1	256	3×3	Max-pooling 2×2	0.2	ReLU
	ConvNet Layer 2	128				
	ConvNet Layer 3	64		Max-pooling 1×1		
ConvNet#4	ConvNet Layer 1	256	3×3	Max-pooling 2×2	0.2	ReLU
	ConvNet Layer 2	128				
	ConvNet Layer 3	128				
	ConvNet Layer 4	64		Max-pooling 1×1		

ConvNet: Convolutional neural network; ReLU: rectified linear unit.

Table 2. ConvNet Experiments and General Properties.

Experiment No.	ConvNet Architecture	Input Dimension	Pre-Processing	Dense Layer #1	Dense Layer #2	Dense Layer #3
Exp.1	ConvNet#1	160×120	Sharpening	128	8	—
Exp.2	ConvNet#2	160×120	Sharpening	128	8	—
Exp.3	ConvNet#3	160×120	Sharpening	128	8	—
Exp.4	ConvNet#4	160×120	Sharpening	128	64	8
Exp.5	ConvNet#1	30×20	Sharpening	128	8	—
Exp.6	ConvNet#2	30×20	Sharpening	128	8	—
Exp.7	ConvNet#3	30×20	Sharpening	128	8	—
Exp.8	ConvNet#1	30×20	APPN	128	8	—
Exp.9	ConvNet#2	30×20	APPN	128	8	—
Exp.10	ConvNet#3	30×20	APPN	128	8	—
Exp.11	ConvNet#1	160×120	—	128	8	—
Exp.12	ConvNet#2	160×120	—	128	8	—
Exp.13	ConvNet#3	160×120	—	128	8	—
Exp.14	ConvNet#4	160×120	—	128	64	8
Exp.15	ConvNet#1	30×20	—	128	8	—
Exp.16	ConvNet#2	30×20	—	128	8	—
Exp.17	ConvNet#3	30×20	—	128	8	—

APPN: Average pixel per node; ConvNet: convolutional neural network.

5.3.1. **Model Statistical Measurement Experiments**

Each image hides basic statistical information that is useful for machine learning models. Consideration number of values instead computational time while achieving reasonable results. In this research, basic statistical information and the pre-processed characteristics were obtained from the images.

An edge esteem was resolved as half of the most extreme pixel esteem inside the picture, and the quantity of pixels more noteworthy and more modest than this worth were tallied. At that point, the picture vertically, and the middle one was the amplest so as the area of interest. The mean estimations portion were determined independently. This cycle was performed to wipe out the corners and outskirts inside the picture. The mean estimations of Laplacian channel, honed picture, and histogram evening out applied pictures were determined Other than these estimations, the base and most extreme pixel esteems inside the picture, picture entropy, standard deviation, change, and the mode were determined. Table 2 shows the made factual and key properties of the pictures in detail. An element vector with 14 ascribes, depicted above, was made and taken care of to five AI classifiers: SVM, LR, nB, DT, and KNN.

CHAPTER 6

6. RELATED WORK

6.1. FOOD RECOGNITION BASED ON GEOMETRY FEATURES

Food category recognition and analysis has been a popular research area in the _eld of nutrition study. However, it is relative difficult because food items are deformable objects with significant variations in appearance. The food items may either have a high intra-class variance (similar foods such as beef and steak look very different based on how to cook them), or low inter-class variance (different foods like _Shand pork look very similar). Different approaches have been proposed to recognize food items in image using geometry features such as SIFT [7] descriptor, color histograms or GIST , and shape context [9]. Moreover, Felzenszwalb [12] use triangulated polygons to represent a deformable shape for detection and Jiang *et al.* proposes learning a mean shape of the target class based on the thin plate spline parametrization.

Besides, Balgonie [9] choose *n* pixels from the contours of a shape and then form *n/1* vectors as a description of the shape at the pixel level. Though geometry feature-based approaches work well in object detection for the certain types of items, there are two main problems for food related tasks. The rust problem is that geometry feature-based methods need to detect features like edges, counters and key points or landmarks, which may not be available in food images.

The other problem is that it is hard to describe the shape of a food item in real world thus calculating shape similarity is very hard.

6.2. FOOD RECOGNITION BASED ON STATISTICAL FEATURES METHODS

To overcome the problems described above, approaches using statistical features are proposed. Instead of edge or key points, the methods focus on local, statistical features like pairs of pixels. Since the statistical distribution of pairwise local features could extract important shape characteristics and spatial relationships between food ingredients, thus facilitating more

accurate results in object recognition. For example, Yang *et al.* [6] explore the spatial relationships between different ingredients (e.g., vegetables and meat in one meal) by employing a multi-step discriminative classifier. Each pixel in the image is assigned a vector indicating the probability of the pixel belongs to nine food ingredients [5]. A multi-dimensional histogram is generated by using pairwise statistic local features, then the histogram is passed into a multi-class SVM for image classification.

6.3. FOOD RECOGNITION BASED ON MACHINE LEARNING METHODS

Recently, there has been an increasing number of research conducting experiments and researches toward the _elds of food classification , leveraging machine learning/deep learning algorithms. Aizawa *et al.* [5] proposed a Bayesian framework-based approach to facilitates incremental learning for both food detection and food-balance estimation. Bussard *et al.* [11] used Random Forest on the Food-101 test set achieving a classification accuracy with 50.67% by mining discriminative components. The random forest model is used for clustering the super pixels of the training dataset [13]. Other advanced classification techniques were also applied in the work including Improved Fisher Vectors (IFV) [12], Bag-of-Words Histogram (BOW) , Randomized Clustering Forests (RCF) [10] and Mid-Level Discriminative Super pixels (MLDS). As the computational power is getting stronger, convolutional neural network (CNN) and deeper models are also widely used in food recognition and provide better performance. Kaaga *et al.* applied the CNN model in food image classification. They achieved a very high accuracy of 93.8% on food and non-food item detection. The experimental results on food recognition showed that the proposed CNN solution outperformed all other baseline methods - achieved an average accuracy of 73.7% for 10 classes. What is more, a _ne-tuned the AlexNet model is used in the work [13]. The method achieved the promising results on public food image datasets so far, with top-1 accuracy of 67.7% CNN model to evaluate the effectiveness of deep model in food image classification. The model is based on the speciation's of Google's image recognition architecture - Inception. In addition, Google Net was used in [9] for food recognition to build a Im2Calories system on Food-101 dataset.

Additionally, researchers start to investigate which features and models are more suitable for the food recognition, and comply them into food analysis system to calculate the calories. In order to automatically estimates the food calories from a food image, multi-task convolutional neural networks is used for simultaneous learning of food calories, categories, ingredients [1]. What's more, a generative adversarial network approach is also proposed for food image analysis [2].

CHAPTER 7

7. THEORITICAL ANALYSIS

For this paper, the researchers the confusion matrix to conduct statistical analysis for the determination of food to be used in this paper since the true and false values for the results was known. Confusion matrix table describes a recognition performance by using a set of test data having true values

N = 110	Automatic True Recognition	Automatic False Recognition	
Actual True Recognition	101	9	110
	101	9	

.

a framework We used Graph cut segmentation and a deep neural network in this analysis. In contrast to our previous work in [12], the combination of these two approaches helps us to greatly improve the accuracy of our food classification and recognition. We would have the size and shape of the food portions by using these two models, as well as the size and shape of the food portions from the graph cut algorithm., We will have the opportunity to measure the calorie content of whole food portions. The first step in our approach is to create a pre-trained model file using the CNN network before implementing the image recognition algorithm in the Android programme. This phase was completed by first taking a collection of images of a single class (for example, 50 images of the apple class) after which they will be marked. after which they are labelled with object name-set (object being apple). The collection of related (positive) images is comprised of these images. The device is trained with these images after the picture sets are captured. The machine is then retrained using a series of negative images (images that do not contain the relevant objects).

Around half of each group's fruit images are used to train the device, while the remaining images serve as the testing collection. We can improve recognition by using graph cut segmentation and the Deep Neural Network algorithm, as shown in Figure 5 and Table 3. In just 3 seconds, our machine was able to correctly identify the food portions.

Table I indicates that we have a 99 percent accuracy rate in single food portioning. In addition, we use a calculation approach in our work that involves the following: Repeating the same measurement several times and better measuring uncertainties by testing how reproducible the measurements are are two ways to improve confidence in experimental results.

There are three important statistical quantities to remember when dealing with repeated measurements: average or mean (an approximation of the "true" value of the measurement), standard deviation (a measure of the "spread" in the data), and standard error (estimate in the uncertainty in the average of the measurements). There are over 100 images in each group. When dealing with a lot of measurements, it's best to use a spreadsheet.

Food items	Real Calories	Average Calories	Standard Error
Red Apple	80	80	0
Orange	71	70	0
Tomato	30	30	0.01
Carrot	30	28	0.1
Bread	68	68	0.5
Pasta	280	276	0.3
Egg	17	17	0
Banana	10	10	0
Cucumber	30	30	0.25
Green Pepper	16	16	0.04
Strawberry	53	52	0.5

TABLE: items of calories and estimeted calories

Results shows the average calories are so close to the real one and also the small range of standard error also shows the accuracy of the system. The overall accuracy of the system with both methods. the application will prompt the user to enter the correct food type and would further display the estimated calorie value based on user's entered information. In this paper, our data set comprises of 30 different categories of food and fruits. These food and fruit images are divided into training and testing sets, where around 50% of the fruit images from each group are used to train the system and the remaining images serve as the testing set.

shows that the null hypothesis failed to be rejected for some items, while for other items, the alternative hypothesis was accepted. The acceptance of alternative hypothesis can be accounted to the limitation of the ultrasonic sensor in getting the accurate height of the food in question.

Moreover, there is a possibility that the density obtained by the researches can be erroneous. Moreover, the algorithm used may not be powerful enough to accurately measure the area, most especially when the food has black regions.

	Recognition Rate (%)			
	Food items	**Using color-texture segmentation**	**Using graph-cut, color-texture segmentation**	**Using Deep Neural Network Method**
1	Red Apple	97.64	100	100
2	Orange	95.59	97.5	99
3	Corn	94.85	96	99.5
4	Tomato	89.56	95	100
5	Carrot	99.79	100	100
6	Bread	98.39	99	99
7	Pasta	94.75	98	100
8	Sauce	88.78	92	98
9	Chicken	86.55	89	100
10	Egg	77.53	83	100
11	Cheese	97.47	97	100
12	Meat	95.73	96	100
13	Onion	89.99	93	99.4
14	Beans	98.68	98	100
15	Fish	77.7	85	100
16	Banana	97.65	97	100
17	Green Apple	97.99	97	99
18	Cucumber	97.65	98	100
19	Lettuce	77.55	85	100
20	Grapes	95.7	95	98
21	Potato	88.56	89	100
22	Tangerine	97.59	99	100
23	Chocolate Cake	88.19	85	100
24	Caramel Cake	85.29	85	100
25	Rice	94.85	94	100
26	Green Pepper	97.99	98	100
27	Strawberry	83.47	98	99
28	Cooked Vegetable	92.62	96	100
29	Cabbage	77.55	100	100
30	Blueberry	83.47	95	100
Total average		92.21	95	99

TABLE: data of which is analised items

CHAPTER 8

8. EXPERIMENTAL INVESTIGATION

code:

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
```

```
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Food</th>\n",
"      <th>Serving</th>\n",
"      <th>Calories</th>\n",
"      <th>Calorie</th>\n",
"      <th>name</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <td>245</td>\n",
"      <td>Corn Oil</td>\n",
"      <td>1 tbsp (15 ml)</td>\n",
"      <td>120 cal</td>\n",
```

```
"      <td>120</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>218</td>\n",
"      <td>Pretzel Sticks</td>\n",
"      <td>20 sticks (12 g)</td>\n",
"      <td>46 cal</td>\n",
"      <td>46</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>131</td>\n",
"      <td>Chicken Fried Steak</td>\n",
"      <td>1 steak (280 g)</td>\n",
"      <td>423 cal</td>\n",
"      <td>423</td>\n",
"      <td>cal</td>\n",
"    <tr>\n",
"      <td>268</td>\n",
"      <td>Vegetable Oil</td>\n",
"      <td>1 tbsp (15 ml)</td>\n",
```

```
"    <tr>\n",
"      <td>352</td>\n",
"      <td>McFlurry</td>\n",
"      <td>1 mcflurry (134 g)</td>\n",
"      <td>205 cal</td>\n",
"      <td>205</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>309</td>\n",
"      <td>Pea Soup</td>\n",
"      <td>1 cup (253 g)</td>\n",
"      <td>190 cal</td>\n",
"      <td>190</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>355</td>\n",
"      <td>Mint Chocolate Chip Ice Cream</td>\n",
"      <td>1 scoop (72 g)</td>\n",
"      <td>172 cal</td>\n",
"      <td>172</td>\n",
```

```
"      <td>627 cal</td>\n",
"      <td>627</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>82</td>\n",
"      <td>Greengage</td>\n",
"      <td>1 fruit (5 g)</td>\n",
"      <td>2 cal</td>\n",
"      <td>2</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>405</td>\n",
"      <td>Whole Grain Noodles</td>\n",
"      <td>2 oz. (56 g)</td>\n",
"      <td>194 cal</td>\n",
"      <td>194</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>529</td>\n",
```

```
"      <td>627 cal</td>\n",
"      <td>627</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>82</td>\n",
"      <td>Greengage</td>\n",
"      <td>1 fruit (5 g)</td>\n",
"      <td>2 cal</td>\n",
"      <td>2</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>405</td>\n",
"      <td>Whole Grain Noodles</td>\n",
"      <td>2 oz. (56 g)</td>\n",
"      <td>194 cal</td>\n",
"      <td>194</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>529</td>\n",
```

```
"      <td>Italian BMT</td>\n",
"      <td>1 sandwich (224 g)</td>\n",
"      <td>410 cal</td>\n",
"      <td>410</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>204</td>\n",
"      <td>Cracker</td>\n",
"      <td>1 cracker (7 g)</td>\n",
"      <td>35 cal</td>\n",
"      <td>35</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <td>248</td>\n",
"      <td>Grape Seed Oil</td>\n",
"      <td>1 tbsp (14 ml)</td>\n",
"      <td>124 cal</td>\n",
"      <td>124</td>\n",
"      <td>cal</td>\n",
"    </tr>\n",
```

```
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"                           Food            Serving Calories  Calorie name\n",
"245                    Corn Oil      1 tbsp (15 ml)  120 cal      120  cal\n",
"218                Pretzel Sticks    20 sticks (12 g)   46 cal       46  cal\n",
"131           Chicken Fried Steak     1 steak (280 g)  423 cal      423  cal\n",
"268                 Vegetable Oil      1 tbsp (15 ml)  120 cal      120  cal\n",
"351                  Magnum White     1 magnum (86 g)  255 cal      255  cal\n",
"93                     Mulberries       1 cup (140 g)   60 cal       60  cal\n",
"55                         Squash    1 squash (196 g)   88 cal       88  cal\n",
"182                  Sausage Roll      1 roll (103 g)  361 cal      361  cal\n",
"370                   Cappelletti        2 oz. (56 g)   92 cal       92  cal\n",
"352                      McFlurry  1 mcflurry (134 g)  205 cal      205  cal\n",
"309                      Pea Soup       1 cup (253 g)  190 cal      190  cal\n",
"355  Mint Chocolate Chip Ice Cream      1 scoop (72 g)  172 cal      172  cal\n",
"446                    Tuna Pizza     1 slice (125 g)  318 cal      318  cal\n",
"178               Reuben Sandwich  1 sandwich (308 g)  641 cal      641  cal\n",
"210                         Kamut       1 cup (186 g)  627 cal      627  cal\n",
"82                      Greengage       1 fruit (5 g)    2 cal        2  cal\n",
```

```
     "405           Whole Grain Noodles       2 oz. (56 g)  194 cal      194  cal\n",
     "529                    Italian BMT  1 sandwich (224 g)  410 cal      410  cal\n",
     "204                        Cracker     1 cracker (7 g)   35 cal       35  cal\n",
     "248                 Grape Seed Oil      1 tbsp (14 ml)  124 cal      124  cal"
    ]
   },
   "execution_count": 2,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "\n",
  "import pandas as pd\n",
  "from sklearn.utils import shuffle\n",
  "df=pd.read_csv('Food and Calories.csv')\n",
  "df[['Calorie','name']] = df.Calories.str.split(expand=True) \n",
  "df['Calorie']=pd.to_numeric(df[\"Calorie\"])\n",
  "df.sort_values(by=['Calorie'], inplace=True,ascending=False)\n",
  "df= shuffle(df)\n",
  "df.head(20)"    ]
}, {
```

"cell_type": "code",

"execution_count": null,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"M\n",

"['Corn Oil 1 tbsp (15 ml)', 'Pretzel Sticks 20 sticks (12 g)', 'Chicken Fried Steak 1 steak (280 g)', 'Vegetable Oil 1 tbsp (15 ml)', 'Magnum White 1 magnum (86 g)', 'Mulberries 1 cup (140 g)', 'Squash 1 squash (196 g)', 'Sausage Roll 1 roll (103 g)', 'Cappelletti 2 oz. (56 g)', 'McFlurry 1 mcflurry (134 g)']\n",

"M\n",

"['Corn Oil 1 tbsp (15 ml)', 'Pretzel Sticks 20 sticks (12 g)', 'Chicken Fried Steak 1 steak (280 g)', 'Vegetable Oil 1 tbsp (15 ml)', 'Magnum White 1 magnum (86 g)', 'Mulberries 1 cup (140 g)', 'Squash 1 squash (196 g)', 'Sausage Roll 1 roll (103 g)', 'Cappelletti 2 oz. (56 g)', 'McFlurry 1 mcflurry (134 g)', 'Pea Soup 1 cup (253 g)']\n",

"M\n",

"['Corn Oil 1 tbsp (15 ml)', 'Pretzel Sticks 20 sticks (12 g)', 'Chicken Fried Steak 1 steak (280 g)', 'Vegetable Oil 1 tbsp (15 ml)', 'Magnum White 1 magnum (86 g)', 'Mulberries 1 cup (140 g)', 'Squash 1 squash (196 g)', 'Sausage Roll 1 roll (103 g)', 'Cappelletti 2 oz. (56 g)', 'McFlurry 1 mcflurry (134 g)', 'Pea Soup 1 cup (253 g)']\n",

"M\n",

```
"['Corn Oil 1 tbsp (15 ml)', 'Pretzel Sticks 20 sticks (12 g)', 'Chicken Fried Steak 1 steak (280 g)', 'Vegetable Oil 1 tbsp (15 ml)', 'Magnum White 1 magnum (86 g)', 'Mulberries 1 cup (140 g)', 'Squash 1 squash (196 g)', 'Sausage Roll 1 roll (103 g)']\n"
     ]    }   ],
   "source": [
    "import tkinter\n",
    "import tk_tools\n",
    "import pandas as pd\n",
    "\n",
    "tk = tkinter.Tk()\n",
    "tk.title ('Calorie calculator')\n",
    "tk.geometry('600x700')\n",
    "\n",
    "Bmi1 = tkinter.StringVar()\n",
    "Bmi2 = tkinter.StringVar()\n",
    "Bmi3 = tkinter.StringVar()\n",
    "\n",
    "label_height = tkinter.Label (tk, text = 'height', font = ('Lishu', 20))\n",
    "label_height.place(x = 10,y = 10,width = 80,height = 40)\n",
    "\n",
    "entry_height = tkinter.Entry(tk,textvariable = tkinter.StringVar(),\n",
    "font = ('Lishu', 20))\n",
    "entry_height.place(x = 90,y = 10,width = 80,height = 40)\n",
```

```
"\n",
"label_cm = tkinter.Label (tk, text = 'cm', font = ('Lishu', 20))\n",
"label_cm.place(x = 170,y = 10,width = 40,height = 40)\n",
"\n",
"label_weight = tkinter.Label (tk, text = 'weight', font = ('Lishu', 20))\n",
"label_weight.place(x = 10,y = 60,width = 80,height = 40)\n",
"\n",
"\n",
"entry_weight = tkinter.Entry(tk,textvariable = tkinter.StringVar(),\n",
"font = ('Lishu', 20))\n",
"entry_weight.place(x = 90,y = 60,width = 80,height = 40)\n",
"label_kg = tkinter.Label (tk, text = 'kg', font = ('Lishu', 20))\n",
"label_kg.place(x = 170,y = 60,width = 40,height = 40)\n",
"\n",
"label_age = tkinter.Label (tk, text = 'age', font = ('Lishu', 20))\n",
"label_age.place(x = 10,y = 120,width = 80,height = 40)\n",
"entry_age = tkinter.Entry(tk,textvariable = tkinter.StringVar(),\n",
"font = ('Lishu', 20))\n",
"entry_age.place(x = 90,y = 120,width = 80,height = 40)\n",
"label_year = tkinter.Label (tk, text = 'years', font = ('Lishu', 20))\n",
"label_year.place(x = 170,y = 120,width = 80,height = 40)\n",
"\n",
```

```
"\n",
"label_gender = tkinter.Label (tk, text = 'Gender', font = ('Lishu', 20))\n",
"label_gender.place(x = 40,y = 180,width = 100,height = 40)\n",
"entry_gender = tkinter.Entry(tk,textvariable = tkinter.StringVar(),\n",
"font = ('Lishu', 20))\n",
"entry_gender.place(x = 180,y = 180,width = 80,height = 40)\n",
"label_gender = tkinter.Label (tk, text = 'M/F', font = ('Lishu', 20))\n",
"label_gender.place(x = 220,y = 180,width = 80,height = 40)\n",
"\n",
"\n",
"def bmi():\n",
"    print(entry_gender.get())\n",
"    if  (str(entry_gender.get()) == \"M\" or  str(entry_gender.get())==\"F\"):\n",
"        w=float(entry_weight.get())\n",
"        h=float(entry_height.get())\n",
"        a=int(entry_age.get())\n",
"        G=str(entry_gender.get())\n",
"        if G=='M':\n",
"            BMR = (10*w )+ (6.25* h )- (5 *a) + 5\n",
"        else:\n",
"            BMR = (10*w )+ (6.25* h )- (5 *a) + 5-161\n",
"        RecCal=('Required Calories is '+str(BMR))\n",
```

```
"\n",
"        \n",
"        \n",
"        bmi_set =
round(float(entry_weight.get())/(float(entry_height.get())*float(entry_height.get()))*10000,2)\n",
"        if bmi_set < 18.5:\n",
"            \n",
"            result = ('Your BMI is:'+str(bmi_set))\n",
"            abc = ('too light')\n",
"        elif 18.5 <= bmi_set <= 25:\n",
"            result = ('Your BMI is:'+str(bmi_set))\n",
"            abc = ('normal')\n",
"        elif 25 <= bmi_set <= 28:\n",
"            result = ('Your BMI is:'+str(bmi_set))\n",
"            abc = ('Overweight')\n",
"        elif 28 <= bmi_set <= 32:\n",
"            result = ('Your BMI is:'+str(bmi_set))\n",
"            abc = ('obesity')\n",
"        else:\n",
"            result = ('Your BMI is:'+str(bmi_set))\n",
"            abc = ('Severe Obesity')\n",
"        \n",
"        Bmi1.set(result)\n",
```

```
"        Bmi2.set(abc)\n",
"        Bmi3.set(RecCal)\n",
"        p=int(bmi_set)\n",
"    else:\n",
"        result = ('Please Select M/F ')\n",
"        abc = ('oops!')\n",
"        Bmi1.set(result)\n",
"        Bmi2.set(abc)\n",
"        Bmi3.set('Ooops!')\n",
"        p=0\n",
"    \n",
"    \n",
"    k=0\n",
"    p=0\n",
"    bmr=int(BMR)\n",
"    l=[]\n",
"    for i,j in df.iterrows():\n",
"        if k>bmr:\n",
"            break\n",
"        else:\n",
"            if(j['Calorie']<=bmr):\n",
"                k=k+int(j['Calorie'])\n",
```

```
"            l.append(str(j['Food'])+\" \"+str(j['Serving']))\n",
"            p=p+1\n",
"    \n",
"    print(l)\n",
"    \n",
"    myFrame = tkinter.Frame(tk).place(x=60, y=450)\n",
"    for i in l:\n",
"        tkinter.Label(myFrame, text = \"● \"+i).pack()\n",
"\n",
"    \n",
"    \n",
"    \n",
"    gauge = tk_tools.Gauge(tk, max_value=100.0,\n",
"                    label='Cal', unit='Cal')\n",
"    gauge.place(x = 100,y = 450,width = 340,height = 250)\n",
"    gauge.set_value(p)\n",
"        \n",
"\n",
"button_bmi = tkinter.Button (tk, text = 'Caluclate', font = ('Lishu', 20),\n",
"command = bmi)\n",
"button_bmi.place(x = 50,y = 230,width = 300,height = 40)\n",
"\n",
```

```
   "entry_bmi1 = tkinter.Entry (tk, textvariable = Bmi1, font = (\"Lishu\", 20))\n",
   "entry_bmi1.place(x = 30,y = 280,width = 340,height = 50)\n",
   "entry_bmi2 = tkinter.Entry (tk, textvariable = Bmi2, font = (\"Lishu\", 20))\n",
   "entry_bmi2.place(x = 30,y = 320,width = 340,height = 50)\n",
   "entry_bmi3 = tkinter.Entry (tk, textvariable = Bmi3, font = (\"Lishu\", 20))\n",
   "entry_bmi3.place(x = 30,y = 380,width = 340,height = 50)\n",
   "\n",
   "\n",
   "\n",
   "tk.mainloop()"
  ]   },  {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
 },  {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
```

```
  }  ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.7.4"
  }  },
 "nbformat": 4,
 "nbformat_minor": 2
}
```

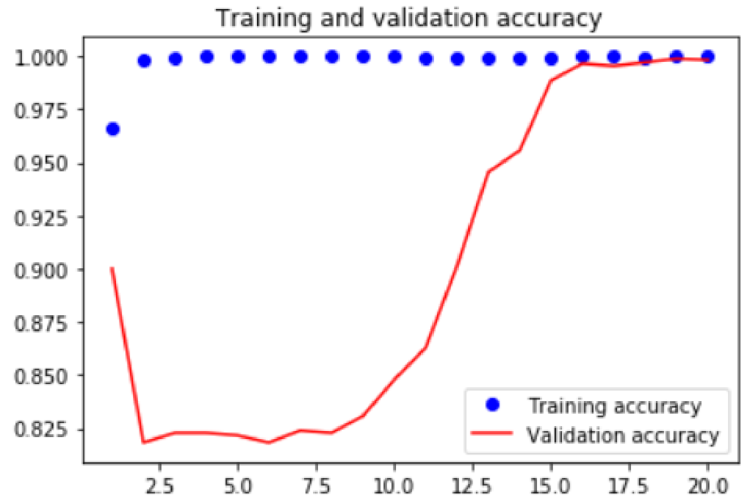
CHAPTER 9

9. RESULTS

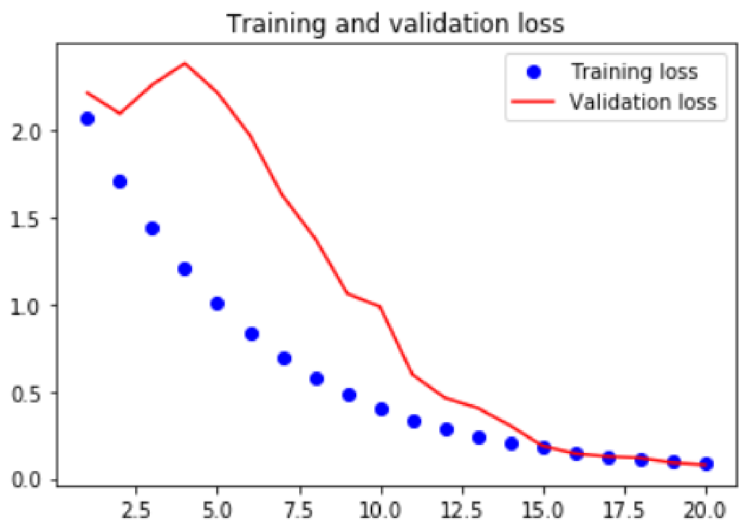
NO	Food Name	Training	Testing	total	calories	Estimated calories
1	Apple	80	20	100	53.96	40.48
2	Banana	88	20	108	170.88	188.81
3	Carrot	352	20	372	31.16	26.28
4	Cucumber	116	20	136	29.44	37.65
5	Onions	122	20	142	44.88	37.13
6	Orange	241	21	262	69.09	71.92
7	Tomato	100	20	120	17.46	13.82
8	Rice	128	20	148	56.45	54.23
9	cheese	293	20	313	65.23	68.52

Table.1: Training and Testing data set for quality Analysis

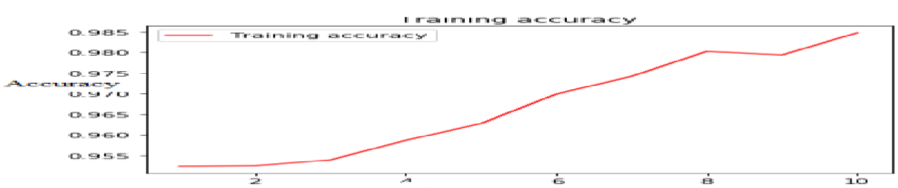
The customer sends a picture location solicitation and cycles it on the worker side. We actualized three classifier SVM, ANN, CNN for analyse the improved precision of framework. counting a pre-prepared CNN model preparing module for grouping purposes, a book information preparing module for trait assessment models, and a worker side module. We explored different avenues regarding an assortment of food classes, each containing a great many pictures, and through AI preparing to accomplish higher grouping exactness.



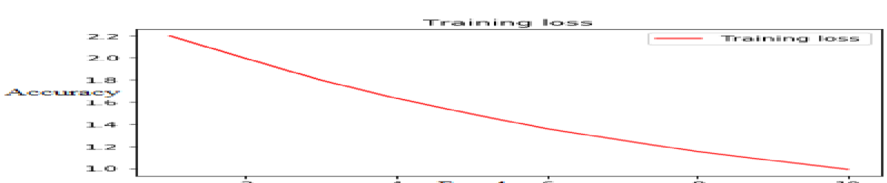
Training and validation accuracy
Training accuracy
Validation accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



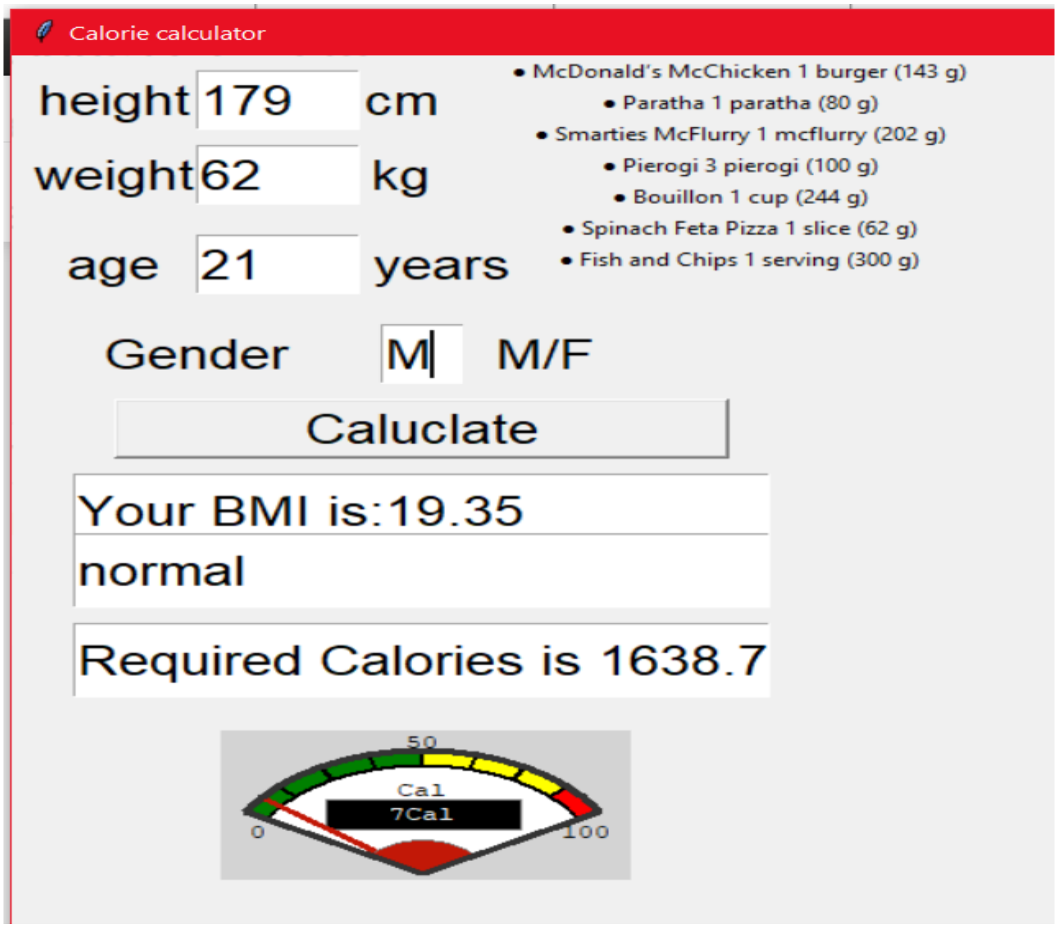
Training and validation loss
Training loss
Validation loss
2.0
1.5
1.0
0.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



Training accuracy
Accuracy
Epoch



Training loss
Training loss
Accuracy
Epoch



CHAPTER 10

10. CONCLUSION

In this the proposed the utilization of an exchange learning and adjusting convolution neural organization model Resnet-50 for picture object discovery. The preparation information of various pictures is utilized to lighten the trouble in portioning food. The neural organization utilized is an adaptable answer for taking care of various scales, sizes, and angle proportions [12]. These issues are significant in visual acknowledgment, however with regards to profound organizations, they frequently get little thought. Additionally, it shows extraordinary precision in division/identification undertakings. Utilizing CNN the testing exactness showed is 97% with a deficiency of 2%. In the current techniques, the precision is less contrasted with CNN

11.REFERENCES:

1. T. Ege and K. Yanai, ``Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions," in *Proc. Thematic Workshops*, 2017, pp. 367_375.
2. T. Miyazaki, G. C. de Silva, and K. Aizawa, ``Image-based calorie content estimation for dietary assessment," in *Proc. IEEE Int. Symp. Multime- dia (ISM)*, Dec. 2011, pp. 363_368.
3. K. Yanai and Y. Kawano, ``Food image recognition using deep convolutional network with pre-training and _ne-tuning," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun./Jul. 2015.
4. Q. Bojia, *Food Recognition and Nutrition Analysis Using Deep CNNs*. Montreal, QC, Canada: McGill Univ., 2019.
5. S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," Computer Vision and Pattern Recognition, IEEE Conference on, pp. 2249 – 2256, 2010.
6. M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," Computer Vision, European Conference on, pp. 18 – 32, Jan 2000.
7. C. Martin, S. Kaya, and B. Gunturk, "Quantification of food intake using food image analysis," Engineering in Medicine and Biology Society.Annual International Conference of the IEEE, pp. 6869 – 6872, 2009.
8. W. Wu and J. Yang, "Fast food recognition from videos of eating for calorie estimation," Multimedia and Expo, IEEE international Conference on, pp. 1210 – 1213, Jan 2009.
9. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. Mougiakkou, "A food recognition system for diabetic patients based on an optimized bag of features model," IEEE Journal of Biomedical and Health Informatics, 2014.
10. L. Bossard, M. Guillaumin, and L. Van Gool, ``Food-101_mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 446_461
11. Y. Kawano and K. Yanai, ``Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. ECCV Workshop Transferring Adapting Source Knowl. Comput. Vis. (TASK-CV)*,2014, pp. 3_17.
12. K. Yanai and Y. Kawano, ``Food image recognition using deep convolutional network with pre-training and _ne-tuning," in *Proc. IEEE Int. Conf.Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1_6.
13. Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi, **"Understanding of a convolutional neural network",** *Department of Computer Engineering, Istanbul Kemerburgaz University, Istanbul, Turkey, August 2017.*