**Chapter 1**

**INTRODUCTION**

**1.1. Introduction**

Heavy storage of digital media, often containing both images and videos is an outcome of growing needs facilitated by modern technology. Providing metadata to images, describing the image content can make it easier to find images of interest, but loses its efficiency with respect to time and cost.

CBIR has provided an automated way to retrieve images based on the content or features of the images itself. Having an image consisting of several items in it, a segmentation algorithm could be applied in order to isolate each item and then process each of them individually. The CBIR system simply extracts the content of the query image matches them to contents of the search image. CBIR is defined as a process to find similar picture or pictures in the image database when a query image is given. Often the image is required to resized, to the database standard size. On doing so the query image could be compared in a better way to the retrieved results. Given a picture of an apple, the system should be able to consider all similar images of an apple in the database. This is done by extracting the features of the images such as color, texture and shape. There will be 5-10 images of every food item in the database, and feature vector would be calculated of each of those images. Ultimately, a final average feature vector of the corresponding food item would be generated. These feature vectors are used to compare between the query image and images in the database. A similarity algorithm is used to calculate the degree of similarity between those two images. Images in the database which has similar images features to the query image are considered for further processing. After finalizing the images from the database, and coming on to the result of what the food item is, calorie value would be mapped from the predefined nutrition table.

As people across the world are becoming more focused on their health it is necessary to keep a track of their diet. It is obvious that eating food with high amount of calories can cause several problems to our health. Although the people can record their meal and discuss with doctors or experts, it is not so convenient and they cannot know the amount of calories before the meal. Thus, “Image Based Calorie Calculator” technique would aid the user to know the calorie value of his meal’s ingredients.

**1.2. Aim and Objective**

The aim of the Calorie Calculator project is to use Content Based Image Retrieval technology to identify and segment different types of elements that are in the image captured by the user. It is identified by first segmenting and then processing it individually. Feature vectors extracted from the user image is compared with the images in the database to identify them. The entire process involves several steps including Pre-Processing, Segmentation, Feature Extraction and Calorie Computation. Various transforms are experimented in order to evaluate accuracy of each.

**1.3. Scope**

The scope of our project " Image Based Calorie Calculator " is to use a new approach in order to help the users to find out the calories they consume by merely just clicking the picture of their meal. The approach can be implemented on various platforms using different tools. Around 5-7 food items have been selected including fruits and vegetables. Input images containing food items from the mentioned dataset will be detected. An item not mentioned in the dataset would produce a result saying "Item not present in the database."

**Chapter 2**

**REVIEW OF LITERATURE**

**2.1. Domain Explanation**

The following project comes under the domain of 'Image Processing'. It uses Content Based Image Retrieval approach to calculate the amount of calories of your meal ingredients.

Digital image processing  is the use of computer [algorithms](https://en.wikipedia.org/wiki/Algorithm) to perform [image processing](https://en.wikipedia.org/wiki/Image_processing) on [digital images](https://en.wikipedia.org/wiki/Digital_image). Content-based image retrieval (CBIR), also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) is the application of [computer vision](https://en.wikipedia.org/wiki/Computer_vision) techniques to the [image retrieval](https://en.wikipedia.org/wiki/Image_retrieval) problem, that is, the problem of searching for [digital images](https://en.wikipedia.org/wiki/Digital_image) in large databases. It opposed to the traditional method of assigning meta data to every image in the database.

The most common method for comparing two images in content-based image retrieval (typically an example image and an image from the database) is using an image distance measure. An image distance measure compares the similarity of two images in various dimensions such as color, texture, shape, and others. For example a distance of 0 signifies an exact match with the query, with respect to the dimensions that were considered. As one may intuitively gather, a value greater than 0 indicates various degrees of similarities between the images. Search results then can be sorted based on their distance to the queried image. Many measures of image distance (Similarity Models) have been developed. Extracting the feature vector from the input image we would be able to implement CBIR technique to identify the what the particular food item would be.

**2.2 Existing Solutions**

Many approaches are available in literature for content based image retrieval based on various techniques and choice of desired image components for feature extraction. Various image components like color, texture, shape, semantic image have proven useful for image retrieval methods. Food recognition exposes new challenges to the current pattern recognition literature and stimulates the stemming of novel techniques for general object recognition. Many people have tried different algorithms at various phases to achieve efficiency on this topic.

At the segmentation phase, each image will be separated into various food portions. To have a considerable segmentation of the image they employed color and texture segmentation tools. For each detected food portion, a feature extraction process has to be performed. In this step, various food features including size, shape, color, and texture will be extracted. The extracted features will be sent to the classification step where, using the support vector machine (SVM) scheme, the food portion will be identified. Finally, by identifying the food item and using some nutritional tables, the calorie value of the food will be extracted.

K-means clustering, is employed for segmentation phase. Clustering is a way to separate group of objects. It treats each object as having a location in space. It finds partitions such that objects within each cluster are as close to each other as possible and as far from objects in other as possible. K means clustering requires number of clusters to be partitioned.

Gabor filter-bank is used for texture feature extraction. The texture features are obtained by subjecting each image to a Gabor filtering operation in a window around each pixel.

The next step is classifying the extracted features in order to recognize each food portion. To do so, SVM is used, which is one of the popular techniques used for data classification. A classification task usually involves training and testing data which consist of some data instances. Each instance in the training set contains one class label and several features. The goal of SVM is to produce a model which predicts the target value of data instances in the testing set which are given only by their attributes.

To measure the size of the food inside the dish, two pictures must be taken: one from the top and one from the side, with the user’s thumb placed beside the dish when taking the

picture from the top. The picture from the side can be used to see how deep the food goes, and is needed for measuring the food portions’ volumes. The system, which already has the

dimensions of the user’s thumb, can then use this information to measure the actual area of each food portion from the top picture, and can multiply this area by the depth (from the side

picture) to estimate the volume of food.

And ultimately, the identified food portion is mapped onto a nutrition table from where the calorie values will be extracted and considered for further calculations.

Input Image

Segmentation Phase

(K-means Clustering)

Feature Extraction Phase

(Gabor Filter Bank)

Data Classification Phase

(Support Vector Machine)

Volume Measurement

(Using thumb)

Calorie Computation Phase

(Database Query Processing)

Total Calories

Figure 2.1 Flowchart of Existing System

**Chapter 3**

**ANALYSIS AND REQUIREMENTS**

**3.1 Functional Requirements**

* The system will take an image as the input, by specifying the file path.
* The user will click an image of the meal ingredients in a plate, as whole items which are not overlapping each other, and the system would perform operations on this image only.
* The system would refine the image by removing unwanted noise and shadows.
* Then system would use Connected Components Labelling algorithm, to segment the food items in the image into different elements. Segmentation helps in processing the food items individually.
* Then every food item is resized to 64\*64 for uniformity in processing.
* Then system extracts feature by applying Walsh /Cosine/Haar Transformation and then it identifies the food item.
* Using the result of feature extraction, it is then mapped on to the nutrition table, to get the calorie value of the identified food item.

**3.2 Non-Functional requirements**

1. **Easy access:** The system should be cost effective hence it can be implemented without much hassle.
2. **Accuracy:** Accuracy of the system is how close the output is to the required output.
3. **Usability:** Ease of use and learnability of the system. System should not be complicated, easy to use and user friendly.
4. **Robustness:** It is the ability of the system to cope with errors during execution.
5. **Reliability:** The probability that the system will perform the required function without failure under stated condition for a stated period of time is called reliability.

**3.3 Software and Hardware Requirements**

**3.3.1 Software Requirement**

* Operating System: Windows 7/8/10
* Programming Language: Matlab

**3.3.2 Hardware Requirement**

* Processor : Intel Core 2 Duo or AMD Athlon64X2
* RAM : 4GB
* Memory : 100MB

**Chapter 4**

**DESIGN**

**4.1 Design Consideration**

While designing the system, all the previous developments made in the domain, and the potential researches that can be effectively considered, have been taken into account.

1. Efficiently segmenting the user image since all the food items are together. If not segmented, recognition of every individual item in the image would fail. Hence the algorithm should efficiently be able to separate the food items for the correct recognition.
2. Taking intelligent decisions while recognizing food items that are of same color like when food items are apples, tomatoes, pomegranates. Look alike items in the image, should result to a correct answer solely depending upon user discretion.
3. No false detection of any item. If a particular item is not present in the dataset should produce a result, " Item not present in the database" and not give incorrect results.
4. Maintaining a decent accuracy rate to recognize the food items while balancing it with the amount of time taken to retrieve calorie values.
5. Finally the task of retrieving the correct calorie values of the identified food items by mapping them with the nutrition table. Such a result is sensitive and important to the user.

**Chapter 5**

**IMPLEMENTATION**

**5.1 Proposed System**

The proposed system enables one to calculate calories by just clicking picture of their meal ingredients. System maintains a nutrition table of all the food items considered in the scope of the project. Having clicked an image of the meal ingredients in a plate, the system will enter the pre processing phase where the noise and shadows will be removed to get a better image for further phases. Following it, in segmentation phase, the image will be analyzed to extract various segments of the food items. It is known that without having a good image segmentation mechanism, it is impossible to process the image appropriately. Hence we have employed 'Connected Components Labelling' algorithm to segment the food items.

For each detected food item from the segmentation phase, a feature extraction process is performed. In this phase, we use various transform to calculate the mean feature vector of the food item. This will be accomplished by applying desired transform and then calculating the mean of each column, we get a mean vector of food item. We have already extracted mean feature vectors of the food items in our scope. The extracted mean feature vector is then compared with the values from our database to identify the food item. Then the identified food item is mapped with our nutrition table to get the calorie values.

The proposed system has major phases:

1. **Pre-Processing phase**

This phase refines the user image of the meal ingredients by removing noise. Then it also removes shadow to get a better and well contrasted image to work on.

1. **Segmentation phase:**

This phase segments the image into different elements using Connected Labelling Components algorithm. It makes it easier for the system to work of each food item individually. Each food item is then resized to 64\*64.

1. **Feature extraction phase:**

This phase extracts feature from each food item segmented from the previous phase. It uses Walsh/Haar/Cosine Transform, and then computes mean feature vector of all the food items from the user image. Pre-Built Database for the food items from the scope is used by comparing them with the mean feature vector of the food items and finally they are identified.

1. **Calorie computation phase:**

This phase calculates the calorie for the identified food items. It maps the food items with the nutrition table to get the calorie values.

Input Image

Pre Processing Phase

(Noise Reduction & Shadow Removal)

Segmentation Phase

(Connected Components Labeling Algorithm)

Feature Extraction Phase

(Feature vector generated using Walsh /DCT/Haar Transform)

Calorie Computation Phase

(Database Query processing)

Total Calories

Figure 3.1 Flowchart of Proposed System

**5.2 Pre processing**

The database would contain images of various food items mentioned in the scope in the .jpg format. The input image will be in the same format.

**5.3 Noise Reduction**

Salt-and-pepper noise is a form of [noise](https://en.wikipedia.org/wiki/Image_noise) sometimes seen on images. It presents itself as sparsely occurring white and black [pixels](https://en.wikipedia.org/wiki/Pixel). An effective [noise reduction](https://en.wikipedia.org/wiki/Noise_reduction) method for this type of noise is a [median filter](https://en.wikipedia.org/wiki/Median_filter) or a [morphological filter](https://en.wikipedia.org/wiki/Mathematical_morphology). The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the [median](https://en.wikipedia.org/wiki/Median) of neighbouring entries. The pattern of neighbours is called the "window", which slides, entry by entry, over the entire signal.

**5.4 Shadow Removal**

We have used Standard deviation to remove shadows, if the deviation is too high, it is recognized as shadow and then that pixel is merged with the background pixel. This allows the system to process the image efficiently. With shadows in the image, the results for feature extraction phase can be hampered.

Algorithm:

1. Take the mean RBG value of each pixel
2. Take Euclidean distance of each pixel in 3 different planes.
3. Check the difference of each plane with other two and if all are below an arbitrary constant, then it is merged with the background or else it is element's pixel.



Figure 5.1 Shadow Removal

**5.5 Segmentation**

Segmentation partitions an image into distinct regions containing each pixels with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Meaningful segmentation is the first step from low-level image processing transforming a grayscale or color image into one or more other images to high-level image description in terms of features, objects, and scenes. The success of image analysis depends on reliability of segmentation, but an accurate partitioning of an image is generally a very challenging problem.

Connected components labelling scans an image and groups its [pixels](http://homepages.inf.ed.ac.uk/rbf/HIPR2/pixel.htm) into components based on [pixel connectivity](http://homepages.inf.ed.ac.uk/rbf/HIPR2/connect.htm), i.e. all pixels in a connected component share similar [pixel intensity values](http://homepages.inf.ed.ac.uk/rbf/HIPR2/value.htm) and are in some way connected with each other. Once all groups have been determined, each pixel is labelled with a gray level or a color (color labelling) according to the component it was assigned to. Extracting and labelling of various disjoint and connected components in an image is central to many automated image analysis applications.

.Pixel connectivity is a method typically used in image processing to analyze which pixels

are connected to other pixels in the surrounding neighbourhoods. Two pixels are considered connected to each other if they are adjacent to each other and their values are from the same set of values. A pixel value in a binary image is an element of the set {0, 1}, of which the 0-valued pixels are called background and the 1-valued pixels are called foreground. The two most widely used methods to formulate the adjacency criterion for connectivity are four-connectivity (N4) and eight-connectivity (N8).

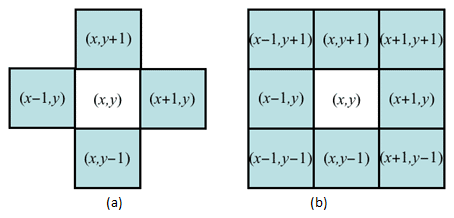


Figure 5.2 (a) 4-Connected (b) 8-Connected

Connected Components Labeling Algorithm:

On the first pass:

1. Iterate through each element of the data by column, then by row (Raster Scanning)
2. If the element is not the background
   1. Get the neighbouring elements of the current element.
   2. If there are no neighbours, uniquely label the current element and continue.
   3. Otherwise, find the neighbour with the smallest label and assign it to the current element.
   4. Store the equivalence between neighbouring labels.

On the second pass:

1. Iterate through each element of the data by column, then by row.
2. If the element is not the background
   1. Re label the element with the lowest equivalent label.

With help of the above mentioned algorithm we will be able to process each of the food portions individually.

****

Figure 5.3 Input Query Image

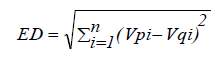
** **

Figure 5.4 (a) Segmented Apple Figure 5.4 (b) Segmented Pomegranate

**5.6 Feature Extraction**

Feature, is the most useful domain of the image. With help of the feature, system will identify what food item it is. To extract feature of the food portions, we are applying Walsh/Cosine/Haar transform and then computing the mean feature vector.

For transforms, feature vectors can be extracted in three ways: applying only column transform, applying only row transform or applying the complete transform. When only column or row transform is employed, row or column mean of the resultant is used as the feature vector for each image respectively. We are applying column transform in our project to get the feature vector of the food item. Absolute value of the coefficients is to be used otherwise they might cancel out each other while taking the mean. The final feature vector is obtained after taking the mean of the feature vectors of all the images in the database and performing energy compaction This energy compaction is to be done by arranging the mean feature vector elements in descending order but maintaining their true positions and considering only the first few coefficients that contribute to 99.5% of the total energy, which is obtained by summing the square of the coefficients. The positions of such coefficients are to be used for finding Euclidean distance during query execution. Here Euclidian distance is used as similarity measure. The direct Euclidian distance between an image P and query image Q can be given as below:

 ---- Equation 5.6.1

where Vp and Vq be the feature vectors of image P and query image Q respectively with size 'n'. Consideration of only those major feature vector coefficients which contribute to most of the energy in the image is reduces the number of feature vector coefficients irrespective of their contribution to the image energy.

Each image of the database resized to 64\*64 while performing row/column feature extraction. In this approach proposed after applying the complete transform, the energy in each of these 10rows is found and averaged over all the images. However, the highest energy coefficient is kept separate in the calculation and always selected in the feature vector. The final feature vector, a 64 row matrix, is obtained.

The technique proposed here splits the frequency domain image i.e. after applying the complete transform for efficient retrieval.

**5.6.1 Discrete Cosine Transform**

A discrete cosine transform (DCT) expresses a finite sequence of [data points](https://en.wikipedia.org/wiki/Data_points) in terms of a sum of [cosine](https://en.wikipedia.org/wiki/Cosine) functions oscillating at different [frequencies](https://en.wikipedia.org/wiki/Frequency). In particular, a DCT is a [Fourier-related transform](https://en.wikipedia.org/wiki/List_of_Fourier-related_transforms) similar to the [discrete Fourier transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) (DFT), but using only [real numbers](https://en.wikipedia.org/wiki/Real_number).

The DCT transform the data from the spatial domain to the frequency domain. The spatial domain can give us an idea about the amplitude of the color as we move through space. The frequency domain explains that the amplitude of the color is varying fast from one pixel to the other pixel in an image data.

The definition of the two dimensional Discrete Cosine Transform can be written as follows, in terms of pixel values f(x, y) for x, y = 0, 1, N-1 and the frequency domain transform coefficient F(u, v),

*F(u, v) = α(u)α(v) ----* Equation 5.6.1

The N x N Cosine Transform Matrix C = {C(u, v)} us defined as

*C(u, v) =* *--* Equation 5.6.2

The 2D DCT of an image can be generated using equation,

*F = C f C’ ------* Equation 5.6.3

Although the direct application of these formulas would require O(*N*2) operations, it is possible to compute the same thing with only O(*N* log *N*) complexity by factorizing the computation similarly to the [fast Fourier transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform) (FFT).

**5.6.2 Walsh Transform**

The Hadamard transform can be regarded as being built out of size-2 [discrete Fourier transforms](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) (DFTs), and is in fact equivalent to a multidimensional DFT of size 2 × 2 × ⋯ × 2 × 2. It decomposes an arbitrary input vector into a superposition of [Walsh functions](https://en.wikipedia.org/wiki/Walsh_function).

The Hadamard transform *Hm* is a 2*m* × 2*m* matrix, the [Hadamard matrix](https://en.wikipedia.org/wiki/Hadamard_matrix) (scaled by a normalization factor), that transforms 2*m*real numbers *xn* into 2*m* real numbers *Xk*. The Hadamard transform can be defined in two ways: [recursively](https://en.wikipedia.org/wiki/Recursively), or by using the[binary](https://en.wikipedia.org/wiki/Binary_numeral_system) ([base](https://en.wikipedia.org/wiki/Base_(exponentiation))-2) representation of the indices *n* and *k*.

Recursively, we define the 1 × 1 Hadamard transform *H*0 by the [identity](https://en.wikipedia.org/wiki/Identity_matrix) *H*0 = 1, and then define *Hm* for *m* > 0 by:

H_m = \frac{1}{\sqrt2} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix} --- Equation 5.6.4

On applying permutation to the Hadamard matrix, the rows of Matrix according the number of sign change of each row. for example:


W(4) = \begin{bmatrix}
1 &  1  & 1 & 1\\
1 & -1  & 1 & -1\\
1 & 1   & -1 & -1\\
1 & -1 & -1  & 1\\
\end{bmatrix}


The successive rows have 0, 3, 1, and 2 sign changes, then we rearrange the rows in ascending ordering.


H(4) = \begin{bmatrix}
1 &  1 &  1 &  1\\
1 &  1 & -1 & -1\\
1 & -1 & -1 &  1\\
1 & -1 &  1 & -1\\
\end{bmatrix}


The Walsh transform can be computed in *n* log *n* operations (*n* = 2*m*), using the [fast Walsh transform](https://en.wikipedia.org/wiki/Fast_Hadamard_transform) algorithm.

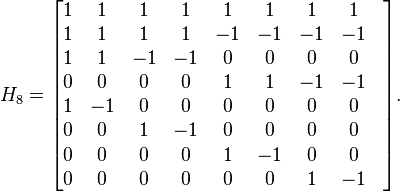
**5.6.3 Haar Transform**

The Haar transform is the simplest of the wavelet transforms. This function against the Haar wavelet with various shifts and stretches, like the Fourier transform cross-multiplies a function against a sine wave with two phases and many stretches.

The Haar wavelet's mother wavelet function \psi(t) can be described as

\psi(t) = \begin{cases}
  1 \quad & 0 \leq  t < \frac{1}{2},\\
 -1 & \frac{1}{2} \leq t < 1,\\
  0 &\mbox{otherwise.}
\end{cases} ---Equation 5.6.5

.

****

 The technical disadvantage of the Haar wavelet is that it is not [continuous](https://en.wikipedia.org/wiki/Continuous_function), and therefore not [differentiable](https://en.wikipedia.org/wiki/Derivative). This property can, however, be an advantage for the analysis of signals with sudden transitions, such as monitoring of tool failure in machines.

**5.7 Calorie Calculation Phase**

Having identified what the food item is by the aid of feature extraction phase, an output string would have the name of that food item. This string would then be searched in the pre-built nutrition table and the corresponding calorie value would be added to total calorie value of your entire meal ingredients.

|  |  |
| --- | --- |
| **Food Items** | **Calorie Value (in kcals)** |
| Apple | 130 |
| Tomato | 30 |
| Potato | 110 |
| Banana | 80 |
| Bread | 250 |
| Pomegranate | 100 |
| Orange | 80 |
| Green Capsicum | 40 |
| Papaya | 55 |

Table 5.1 Nutrition Table

The above table consists of few of the food items we have considered in our scope.

For instance, the meal image consists of 4 items, one apple, one bread, potato and a pomegranate would calculate the total calorie value of the meal as 375(in kcals).



Figure 5.5 Input Query Image

**Chapter 6**

**EXPERIMENTAL RESULTS**

In order to check the accuracy of the proposed solution a data set of images has been prepared by us for the food items which have considered in our scope. Resizing of the images to 256\*256 and 64\*64 is done to maintain the consistency of the data set.

For each image various transforms have been applied and the mean feature vector is compared with each of the feature vector of the items in the database and using Euclidean distance the closeness of the query object is computed against the dataset items. The smallest distance signifies the query object is that particular item,



Figure 6.1 (a) Input Query Image

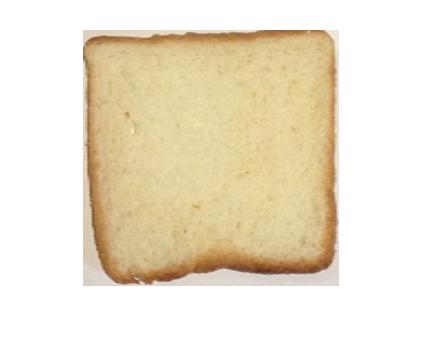


Figure 6.1 (b) Segmented Image of Bread



Figure 6.1(c) Segmented Image of Potato



Figure 6.1 (d) Segmented Image of Pomegranate



Figure 6.1 (d) Segmented Image of Apple

Now the three mentioned transforms are applied against the query image and compared with the existing data set to calculate the accuracy.

The accuracy percentage has been calculated using equation (4) and equation (5).

(4)

(5)

The results for each figure have been shown in the tables below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Walsh** | **Training** | **Testing** | **Error** | **Accuracy** |
| **Apple** | 4237 | 3801 | 0.114707 | **88.52933** |
| **Pomegranate** | 4665 | 3374 | 0.382632 | **61.73681** |
| **Bread** | 2943 | 2129 | 0.382339 | **61.76609** |
| **Potato** | 3798 | 3813 | 0.003934 | **99.60661** |
| **Onion** | 4732 | 3400 | 0.391765 | **60.82353** |

Table 6.1(a) Results of Walsh Transform

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DCT** | **Training** | **Testing** | **Error** | **Accuracy** |
| **Apple** | 3824 | 3799 | 0.006581 | **99.34193** |
| **Pomegranate** | 4043 | 3176 | 0.272985 | **72.70151** |
| **Bread** | 2812 | 2078 | 0.353224 | **64.67757** |
| **Potato** | 3460 | 3363 | 0.028843 | **97.11567** |
| **Onion** | 2742 | 2629 | 0.042982 | **95.70179** |

Table 6.1(b) Results of Discrete Cosine Transform

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Haar** | **Training** | **Testing** | **Error** | **Accuracy** |
| **Apple** | 147 | 137 | 0.072993 | **92.70073** |
| **Pomegranate** | 193 | 191 | 0.010471 | **98.95288** |
| **Bread** | 147 | 140 | 0.05 | **95** |
| **Potato** | 173 | 170 | 0.017647 | **98.23529** |
| **Onion** | 201 | 167 | 0.203593 | **79.64072** |

Table 6.1(c) Results of Haar Transform

The accuracy over the considered data set is has been the most for Haar transform. Whereas with respect to a particular food item the preference may vary. However, the Haar transform has been more than 90% accurate for atleast four of the items. We can easily conclude that applying Haar transform would produce better and accurate results.

**Chapter 7**

**CONCLUSION**

**7.1 Conclusion**

The above discussed method can be used to detect different food items in a particular image by isolating each food item and identifying them. Here, connected components is used to isolate each food item, which also resizes the each item into 64x64 bit image for feature extraction. For the identification of food items, instead of using one single transform, we have used different transforms and checked for accuracy. The accuracy of Haar Transform, when tested on a dataset of images of varying items, is observed to be the highest. This method detects the food items, even if the image has noise and shadows like different brightness, blurriness etc..

**7.2 Future Work**

Future work for this methodology includes implementing the methodology with an additional ability of measuring volume of the food items, thereby increasing the accuracy of calorie computation. This method can be adopted by using an object for which volume is defined universally as a calibration reference for volume measurement of food from the captured photo. Example of such an object can be a coin, for which volume is defined universally.

**REFERENCES**

[1] Sondos M. Fadl, Noura A. Semary, and Mohiy M. Hadhoud: “Fan Search for Image Copy-Move Forgery Detection”, Second International Conference, AMLTA 2014, Cairo, Egypt, November 2014. pp 177-186.

[2] M. Sridevi, C. Mala, S. Sandeep: “Copy-Move Image Forgery Detection In a Parallel Environment”, CS & IT- CSCP 2012, pp. 19-29.

[3] Y. Huang, W. Lu, W. Sun and D. Long: "Improved DCT-based detection of copy-move forgery in images." Elsevier Forensic science international, vol. 206, pp. 178-184, no. 1, 2011.

[4] A. C. Popescu and H. Farid: “Exposing digital forgeries by detectingduplicated image regions." Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2004-515, 2004.

[5] G. Lynch, F. Y.Shih and H.-Y. M. Liao, "An efficient expanding block algorithm for image copy-move forgery detection", Elsevier Science Inc. 2013, pp. 253-265.

[6] Jing Zhang, ZhanleiFeng, Yuting Su: “A new approach for detecting Copy- Move forgery in Digital images”, ICCS 2008, pp. 362-366.

[7] Jie Hu, Huaxiong Zhang, QiangGao, Hai Huang:“An improved lexicographical sort algorithm of Copy –Move Forgery Detection.

[8] Yang Wang, KaitlynGurule, Jacqeline Wise, Jun Zheng: “Wavelet Based Region Duplication Forgery Detection, 2012 Ninth International Conference on Information Technology –New Generations, pp. 30- 35.

[9] Tanuja K. Sarode, Naveen Vaswani: “Region Duplication Forgery Detection using Hybrid Wavelet Transforms”,International Journal of Computer Applications (0975 – 8887) Volume 90 – No 11, March 2014

[10] H. Huang, W. Guo, and Y. Zhang:“Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm,” in Proceedings of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Vol. 2, pp. 272-276, 2008.

[11] H.B. Kekre, ArchanaAthawale, DipaliSadawarti: “Algorithm to Generate Wavelet Transform from an Orthogonal Transform”, International Journal of Image Processing (IJIP), Vol.4, Issue 4, 2010.

[12] H.B. Kekre, Archana Athawale, Dipali Sadavarti: “Algorithm to Generate Kekre’s Wavelet Transform from Kekre’s Transform”, International Journal of Engineering Science and Technology (IJEST), Vol. 2, No. 11, 2010, pp. 756-767.

[13] Dr. H.B. Kekre, Dr. Tanuja K. Sarode, SudeepThepade, Ms. SonalShroff: “Instigation of Orthogonal Wavelet Transforms using Walsh, Cosine, Hartley, Kekre Transforms and their use in Image Compression”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 6, 2011, pp. 125-133.

[14] Dr. H.B. Kekre, Dr. Tanuja K. Sarode, SudeepThepade:

“Inception of Hybrid Wavelet using Two Orthogonal Transforms and its use in Image Compression”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 6, 2011, pp. 80-87.

[15] Image Manipulation Dataset, Unmodified/original images,https://www5.cs.fau.de/fileadmin/research/datasets/image\_forensics\_dataset/forensics\_database/precomputed/orig\_sd.zip, accessed February 2016

[16] Old Volkswagen Rusty Van,http://www.freepik.com/free-photo/old-volkwsagen-rusty-van\_768803.htm, accessed February 2016

[17] Sitting on the pier, http://www.freepik.com/free-photo/sitting-on-the-pier\_773436.htm, accessed February 2016

[18] Clone Detection – elsamuko, https://farm5.static.flickr.com/4094/4869570724\_5439aef726\_b.jpg, accessed February 2016

[19] Keyboard Detail, http://www.freepik.com/free-photo/keyboard-detail\_766381.htm, accessed February 2016