

Object Recognition by Feature Weighted Matrix – A Novel Approach

K. S. Sivaraman¹, S. Gautam¹, S. Sarvesh¹, Archit Khullar¹, A. Baskar^{2*} and Shriram K. Vasudevan³

¹Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University), Coimbatore, India

²Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University), Coimbatore, India; a_baskar@cb.amrita.edu

³Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University), Coimbatore, India; kv_shriram@cb.amrita.edu

Abstract

Objective: This research aims at formulating a method to categorise a given class of objects by obtaining a weighted matrix computed as explained below. **Methods/Analysis:** The method deployed can be branched into two phases: *Training* and *Testing*. In the first phase, a set of images of the concerned objects are taken. By set of images, one can refer to images of different objects, or different positions of the same object. The features are then, extracted for these input images and stored in the database as vectors. Any computation hence forth, is performed using these vectors. In testing stage, the algorithm uses its knowledge to identify the input image to a specified class. **Findings:** Our method is computationally inexpensive since all the calculations are performed on the basic grounds of matrix operations. This method is not just limited to the domain of object recognition alone. Any real-time entity that can be statistically represented in a vector form can be deployed. All that is required of the application is that the range of vectors is defined so as to obtain the minimum components and maximum components, individually. Once this is obtained, the algorithm will be sufficient to identify any input and will accordingly determine the category to which it belongs. The only challenge identified is that the range of vectors obtained from the input data for various categories must not overlap. That being the case will result in multiple hits or in simpler words, will give an incorrect result. Further work can be implemented on how to make the algorithm independent of this dependency. Also, the algorithm improves the results through various illumination and scaling conditions and this has been discussed in results and analysis section. Even with the existing methods to recognize an object, this algorithm can be combined to categorize or classify objects. **Conclusion/Application:** The proposed algorithm successfully classifies the input image into one of the trained categories by identifying the features followed by computing these obtained features as prescribed the given algorithm.

Keywords: Geometrical Modelling, Object Detection, Object Identification, Object Recognition, Weighted Matrix

1. Introduction

Identifying meaningful region of interest in an image and video is a profound research problem in computer vision. Object recognition¹⁻⁴ is the ability to perceive an object's physical attributes such as regions of disparate spatial description; shape, color, and salient texture pattern. There have been several methods and approaches to formulate object recognition. Some of the available approaches to model objects are geometric shape modeling, appearance

based modeling and local or global feature modeling. One such technique, mentioned under appearance based method, is finding the histogram^{5,6} of intensities of a given image, or a region.

Geometric shape modeling⁷⁻¹¹ refers to the technique of realizing the shape of the object, or its outline, and representing it in a generalized form, more specifically, a mathematical function. Many methods are established to realize the shape of an object, using dominant point detection and polygonal approximation, or assuming

*Author for correspondence

the outline of an object to be a collection of vertices, representing the object as a graph, and finding the most important vertices which lead in the shape of the outline. In our approach boundary of the target is divided using 32 equally spaced horizontal lines along their height. For example, an upright triangle has its width increasing along its height proportionally. Or a circle has its width increasing till it reaches the center, and then, decreases, with the same rate it increased. Any random object, given its outline, can be identified in this manner. This method will be delineated in the proposed work.

In this paper, a novel approach called object recognition by feature weighted Matrix is proposed. Our algorithm recognizes the object in two phases, the feature Weighted Matrix Model will be constructed using feature vector and is estimated by Geometrical modelling and Histogram bin in the first phase and in the second phase target, the objects are identified using Diagonal Rank Matrix.

2. Proposed Work

The proposed method recognizes the target object in two phases; in the first phase, the front and top position of the training image is acquired at a distance 50 cm and 100 cm respectively, and a weighted matrix for training dataset is modelled from the corresponding feature vector (Geometric modelling and Histogram bin) and feature matrix for test image is estimated. In the second phase Diagonal Rank Matrix (DRM) is made to discern the category of the target object. The detailed overview of the proposed method is illustrated in Figure 1.

2.1 Training Data Set

In this work, a commonly used home based rigid object is considered as an input. The process of collecting the training data set is as follows: The front and top view of training image is acquired at the distance 50 cm and 100 cm respectively, and for each view and distance, there are five different shots of the same object being generated. A similar procedure can be extended to n objects to obtain a considerable input data set.

2.2 Feature Extraction

Feature extraction is the first phase of the proposed work; the acquired training dataset is fed into two feature extraction algorithms, Geometrical modelling and

Histogram bin. Geometric modelling uses classical edge detection algorithms in pre-processing to detect boundary of the target. The detected boundary is divided using 32 equally spaced horizontal lines $((y_{max}-y_{min})/33)$ along with their height as illustrated in figure 2a-2c. The length of each line $\langle g(u, 1), g(u, 2) \dots g(u, 32) \rangle$ is used as a vector component. Eg_u is the corresponding feature vector of Geometric modelling, as described in the equation below.

$$Eg_u = [g(u, 1) g(u, 2) g(u, 3) \dots g(u, 32)]$$

Where,

$$u \in [1, 5]$$

Histogram bin is a type of histogram that represents the tonal distribution of object by plotting a relation between the number of pixel values and its corresponding intensity by grouping them into bins (32 bins, as proposed), and Feature vector is estimated. The below equation describes the feature vector for the Histogram bin (Eh_u). The figure 3a shows a sample input image and the corresponding Histogram bin shown in the following figure 3b.

$$Eh_u = [h(u, 1) h(u, 2) h(u, 3) \dots h(u, 32)]$$

Where,

$$u \in [1, 5]$$

2.3 Weighted Matrix

The feature vectors (Eg_u , Eh_u) are fed into the newly introduced novel model called Weighted Matrix (S)_{kk} ($k = 4n$). The weighted matrix estimates the likelihood, for an input image, of belonging to the correct category. Further the DRM is generated, which highlights the diagonal elements, comprising of the weighted likelihoods, observing which, the likelihoods can be ranked. The weighted matrix is obtained by processing the Feature vector through six phases, namely:

- Minimum matrix (m)
- Maximum Matrix (M)
- Input Augmentation (Calculating matrix T from vector t)
- Finding Max deviation matrix (R)
- Finding Min Deviation matrix (r)
- Weighted matrix Estimation (S)

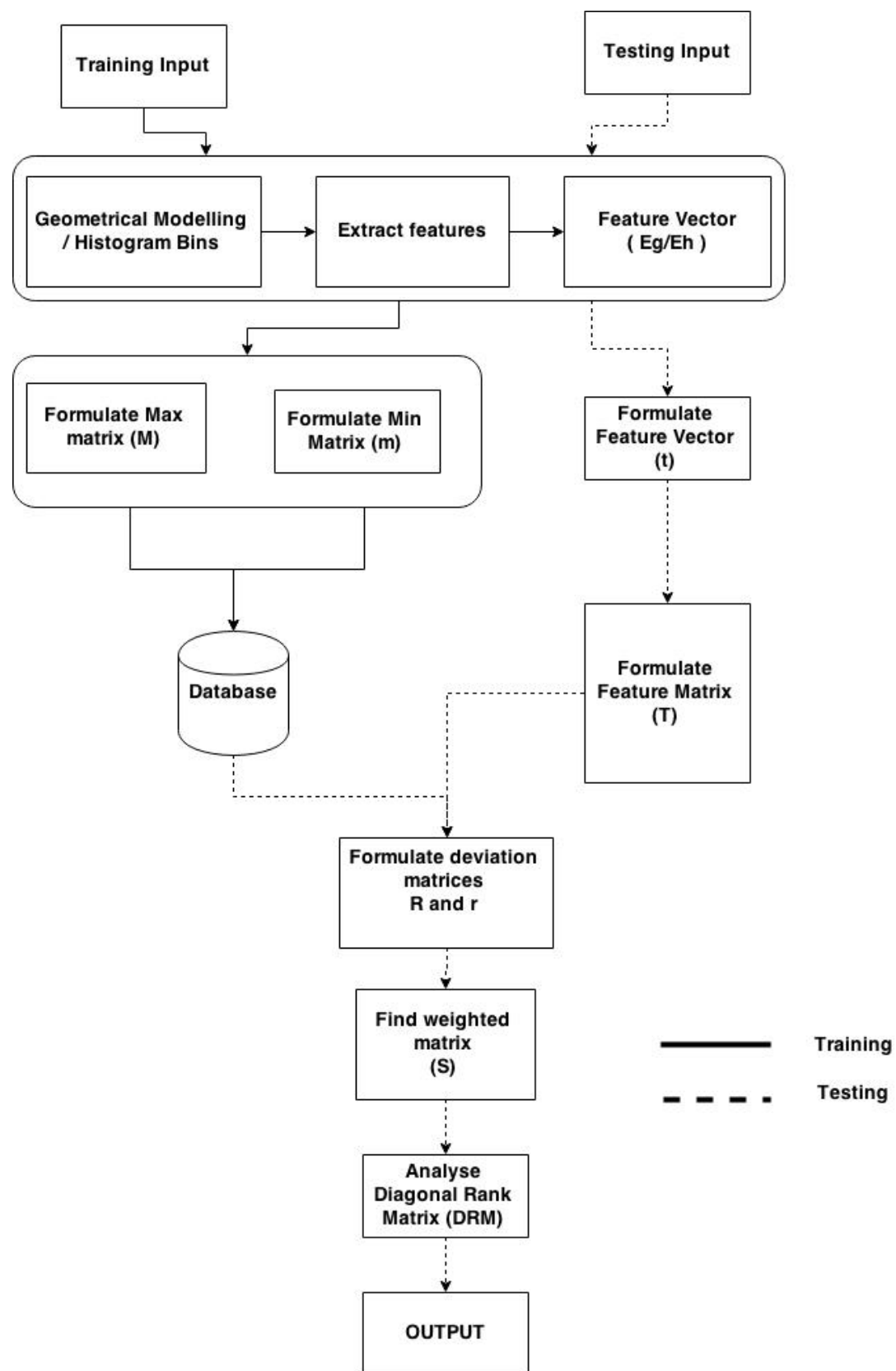


Figure 1. Block diagram for proposed work.



Figure 2a. Sample Input; **Figure 2b.** Boundary of the **Figure 2c.** 32 equally spaced Image Horizontal Line Object.

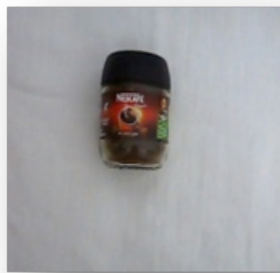


Figure 3a. Sample input.

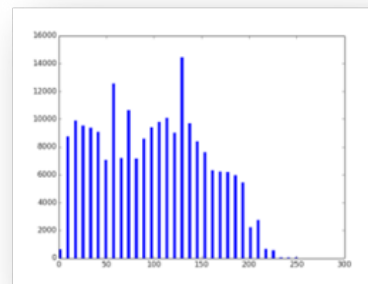


Figure 3b. Histogram bin.

2.3.1 The Minimum Matrix

Let $m_{(1,j)}$, $m_{(2,j)}$, $m_{(3,j)}$ and $m_{(4,j)}$ be the corresponding minimum vectors, where j varies from 1 to 32. The Minimum Matrix can be obtained by finding the minimum of all the components, along the columns from the input dataset, which consists of feature vector obtained for five images of an object in the front view at a distance of 50 cm, five images at a distance of 100 cm, five images taken from the top view at a distance of 50 cm, and another five at a distance of 100cm. The steps are illustrated below using an intermediate matrices.

Minimum matrix for Geometrical modelling:

Minimum vector $m(1, j)$ is described below for the front view of an object is placed at a distance 50 cm

$$m(1, j) = \text{Min} \begin{bmatrix} g(1, j) \\ g(2, j) \\ g(3, j) \\ g(4, j) \\ g(5, j) \end{bmatrix}$$

Minimum vector $m(2, j)$ is described below for the top view of the object is placed at a distance 50 cm

$$m(2, j) = \text{Min} \begin{bmatrix} g(6, j) \\ g(7, j) \\ g(8, j) \\ g(9, j) \\ g(10, j) \end{bmatrix}$$

Minimum vector $m(3, j)$ is described below for the front view of the object is placed at a distance 100 cm

$$m(3, j) = \text{Min} \begin{bmatrix} g(11, j) \\ g(12, j) \\ g(13, j) \\ g(14, j) \\ g(15, j) \end{bmatrix}$$

Minimum vector $m(4, j)$ is described below for the top view of the object is placed at a distance 100 cm

$$m(4, j) = \text{Min} \begin{bmatrix} g(16, j) \\ g(17, j) \\ g(18, j) \\ g(19, j) \\ g(20, j) \end{bmatrix}$$

Where,

$g(1, j) \dots g(20, j)$ Feature vector of input data set

$$j \in [1, 32]$$

Minimum matrix for histogram bin:

Minimum vector $m(1, j)$ is described below for the front view of an object is placed at a distance 50 cm

$$m(1, j) = \text{Min} \begin{bmatrix} h(1, j) \\ h(2, j) \\ h(3, j) \\ h(4, j) \\ h(5, j) \end{bmatrix}$$

Minimum vector $m(2, j)$ is described below for the top View of the object is placed at a distance 50 cm

$$m(2, j) = \text{Min} \begin{bmatrix} h(6, j) \\ h(7, j) \\ h(8, j) \\ h(9, j) \\ h(10, j) \end{bmatrix}$$

Minimum vector $m(3, j)$ is described below for the front View of the object is placed at a distance 100 cm

$$m(3, j) = \text{Min} \begin{bmatrix} h(11, j) \\ h(12, j) \\ h(13, j) \\ h(14, j) \\ h(15, j) \end{bmatrix}$$

Minimum vector $m(4, j)$ is described below for the top View of the object is placed at a distance 100 cm

$$m(4, j) = \text{Min} \begin{bmatrix} h(16, j) \\ h(17, j) \\ h(18, j) \\ h(19, j) \\ h(20, j) \end{bmatrix}$$

Where,

$h(1, j) \dots h(20, j)$ Feature vector of input data set

$$j \in [1, 32]$$

The minimum vectors described above are combined along the column. The minimum matrix for geometrical modelling (mg) and histogram bin (mh) is obtained as follows.

$$mg = \begin{bmatrix} m(1,1) & \dots & m(1,32) \\ \vdots & \ddots & \vdots \\ m(4,2) & \dots & m(4,32) \end{bmatrix}$$

$$mh = \begin{bmatrix} m(1,1) & \dots & m(1,32) \\ \vdots & \ddots & \vdots \\ m(4,2) & \dots & m(4,32) \end{bmatrix}$$

Where,

Size of m_g and m_h is $(4n \times 32)$, n is the number of objects considered.

2.3.2 The Maximum Matrix (M)

The procedure estimating the maximum matrix is similar to that of finding the minimum matrix; in this case we find the maximum of all the components along the column present in the input data set is considered. The steps are illustrated below using an intermediate matrix $M_{(i,j)}$.

Maximum matrix for Geometrical modelling:

Maximum vector $M(1, j)$ is described below for a front view of an object is placed at a distance 50 cm

$$M(1, j) = \text{Max} \begin{bmatrix} l(1, j) \\ l(2, j) \\ l(3, j) \\ l(4, j) \\ l(5, j) \end{bmatrix}$$

Maximum vector $M(2, j)$ is described below for top View of the object is placed at a distance 50 cm

$$M(2, j) = \text{Max} \begin{bmatrix} l(6, j) \\ l(7, j) \\ l(8, j) \\ l(9, j) \\ l(10, j) \end{bmatrix}$$

Maximum vector $M(3, j)$ is described below for front View of the object is placed at a distance 100 cm

$$M(3, j) = \text{Max} \begin{bmatrix} I(11, j) \\ I(12, j) \\ I(13, j) \\ I(14, j) \\ I(15, j) \end{bmatrix}$$

Maximum vector $M(4, j)$ is described below for top View of the object is placed at a distance 100 cm

$$M(4, j) = \text{Max} \begin{bmatrix} I(16, j) \\ I(17, j) \\ I(18, j) \\ I(19, j) \\ I(20, j) \end{bmatrix}$$

Where,

$I(1, j) \dots I(20, j)$ Feature vector of input data set

$$j \in [1, 32]$$

Maximum matrix for histogram bin:

Maximum vector $M(1, j)$ is described below for the front view of an object is placed at a distance 50 cm

$$M(1, j) = \text{Max} \begin{bmatrix} I(1, j) \\ I(2, j) \\ I(3, j) \\ I(4, j) \\ I(5, j) \end{bmatrix}$$

Maximum vector $M(2, j)$ is described below for top view of the object is placed at a distance 50 cm

$$M(2, j) = \text{Max} \begin{bmatrix} I(6, j) \\ I(7, j) \\ I(8, j) \\ I(9, j) \\ I(10, j) \end{bmatrix}$$

Maximum vector $M(3, j)$ is described below for front view of the object is placed at a distance 100 cm

$$M(3, j) = \text{Max} \begin{bmatrix} I(11, j) \\ I(12, j) \\ I(13, j) \\ I(14, j) \\ I(15, j) \end{bmatrix}$$

Maximum vector $M(4, j)$ is described below for top view of the object is placed at a distance 100 cm

$$M(4, j) = \text{Max} \begin{bmatrix} I(16, j) \\ I(17, j) \\ I(18, j) \\ I(19, j) \\ I(20, j) \end{bmatrix}$$

Where,

$I(1, j) \dots I(20, j)$ Feature vector of input data set

$$j \in [1, 32]$$

As done previously for the minimum matrix, the maximum matrix is formed, by combining the maximum vectors along the column of individual vectors. The following matrices M_g and M_h describe the maximum matrix for geometrical modelling and histogram bin, respectively.

$$M_g = \begin{bmatrix} M(1,1) & \dots & M(1,32) \\ \vdots & \ddots & \vdots \\ M(4,2) & \dots & M(4,32) \end{bmatrix}$$

$$M_h = \begin{bmatrix} M(1,1) & \dots & M(1,32) \\ \vdots & \ddots & \vdots \\ M(4,2) & \dots & M(4,32) \end{bmatrix}$$

Where,

Size of m_h and m_g is $(4n \times 32)$,

n is the number of objects considered

2.3.3 Calculating Matrix T from Vector t

Given an input test image, feature vector (t_g and t_h are the feature vector for Geometric model and Histogram bin respectively) is extracted from the feature extraction module. Matrix T_g and T_h obtained from the feature vectors (t_g and t_h) separately by replicating $4n$ times along

the row order.^(*) Initially the feature vectors present in the matrix T_g/T_h are assumed to be a true positive object and in the subsequent process objects are recognized using matrix R , r and DRM. The below equation describes the feature vector for test image and followed by matrix T_g and T_h respectively.

(*) For the reader's understanding, T_g is obtained by replicating t_g $4n$ times, so as to match the rules of matrix subtraction. Since, the minimum and maximum matrices will be of the order $4n \times 32$, the input matrix shall be augmented as mentioned above.

$$tg = [11 \ 12 \ \dots \ 132]$$

$$th = [11 \ 12 \ \dots \ 132]$$

$$Tg = \begin{bmatrix} 11 & \dots & 132 \\ \vdots & \ddots & \vdots \\ 11 & \dots & 132 \end{bmatrix}$$

$$Th = \begin{bmatrix} 11 & \dots & 132 \\ \vdots & \ddots & \vdots \\ 11 & \dots & 132 \end{bmatrix}$$

2.3.4 Deviation from Maximum (R)

The input image feature matrix T_i is subtracted from the maximum matrix M_i . The corresponding deviation matrix, R_i is obtained and described as:

$$Ri = Mi - Ti$$

Where,

T_i stands for Input matrix replicated k times

$$i \in g(\text{or}) h$$

2.3.5 Deviation from Minimum (r)

The minimum matrix m_i is now, subtracted from the input test matrix T_i . Hence, the corresponding deviation matrix r_i is obtained as:

$$ri = Ti - mi$$

Where,

T_i stands for Input matrix replicated k times

$$i \in g(\text{or}) h$$

2.3.6 Formulating the Weighted Matrix ($S_{k \times k}$)

The weighted matrix (S) estimates the likelihood of recognizes the target object and it is obtained by multiplying. The deviation from the maximum R_i matrix and transpose of deviation from the minimum r_i matrix.

$$S = Ri, \text{trans}(ri)$$

Consider a general scenario, where, 'a' and 'b' are two numerical entities, such that 'a' < 'b'. For an input entity, 'x', the sign of the product of (b-x) and (x-a) depends on whether x lies between 'a' and 'b'.

The function given below summarizes:

$$\begin{cases} \text{if } a < x < b & \text{then } (b-x) \text{ and } (x-a) \text{ both positive} \\ \text{if } x < a, & \text{then } (b-x) \text{ is positive and } (x-a) \text{ is negative} \\ \text{if } x > b, & \text{then } (b-x) \text{ is negative and } (x-a) \text{ is positive} \end{cases}$$

Each element in S is a sum of products of the corresponding values in R_i and r_i , respectively. The resultant product of the respective element has remained positive where x lies between 'a' and 'b'. The product would become negative, if one of the elements does not satisfy the condition of 'x' being between 'a' and 'b'. Hence, the components of S that are positive signify that the component from the test vector falls exactly between the corresponding components of the minimum and the maximum matrices. If all the components of a row in S are positive, it is evident that the test input belongs to the category pertaining to the vector specified in that row.

2.3.7 Diagonal Rank Matrix (DRM)

Diagonal rank matrix is the final step of our proposed model; it is used to recognize the object based on ranking, which is estimated by weighted matrix S and then it is scalar multiplied by the Identity(**) matrix I^{4n} . The below equation describes the diagonal rank matrix.

$$\text{DRM} = S \cdot I^{4n}$$

(**) Scalar multiplication is performed on corresponding elements, unlike normal matrix multiplication. The multiplication is performed, without any subsequent addition. In this case, the diagonal elements are multiplied by 1, and are thus retained while the other elements (that are of no consequence), multiplied with 0, are nulled.

The diagonal elements in DRM are a result of the sum of products of R_i and the transpose of r_i . The largest positive value among the diagonal elements contributes to the maximum likeliness of the target object.

The diagonal elements in DRM are a result of the sum of products of R_i and the transpose of r_i . As mentioned earlier, whenever either of the components (let a or b) is negative their multiplication will result in a negative value. This negative value would lead to a reduction in the sum and thus a reduced weight and if both the components are positive (let a or b), the positive product contributes to the weight and thereby, increasing the likelihood. The row corresponding to the largest value, hence, pertains to the category. Likewise, the row corresponding to the least value shows the least possibility of the test input to belong to that category.

3. Implementation Details

We have identified commonly used home based objects namely coffee bottle, Ketchup bottle and coffee cup. The first two objects are used as training data set in this work. The different views and distances are summarized in the table 1. The algorithm has been implemented using Open CV and programmed in Python 2.7; and tested on a Pentium i5 processor with 4 GB RAM and the training and testing images were taken using a normal 2 mega

pixel web camera. The figure 4 shows different views and distance of the training data set.

3.1 Geometric Modelling

The figure 5 and 6 illustrates Minimum and Maximum Feature Vector Visualization for Geometrical model of coffee bottle and ketchup bottle, the red line in this figure describes the minimum widths, representing the minimum matrix values of the corresponding categories and the blue line describes the maximum widths for the same. Duly note that this is done for graphical representation for ease of visualising, and by no means is used for any calculation.

3.2 Histogram Bins

The Minimum and Maximum Feature Vector Visualization for Histogram bin of coffee bottle and ketchup bottle is exemplified in figure 7 and 8, the red line represents the minimum value of the corresponding bin of each category. The blue line represents the maximum values for the same. For a given test image to qualify in one of the categories, the input's histogram bin feature vector has

Table 1. Training data set

Object	Distance of 50cm		Distance of 100 cm	
	Front View	Top View	Front View	Top View
Coffee bottle	5	5	5	5
Ketchup bottle	5	5	5	5



Figure 4. Training data set.

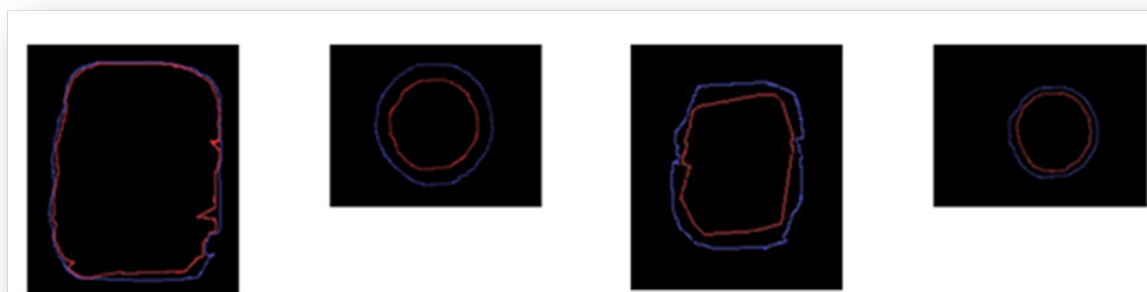


Figure 5. Minimum and maximum feature vector visualization for geometrical model to coffee bottle.

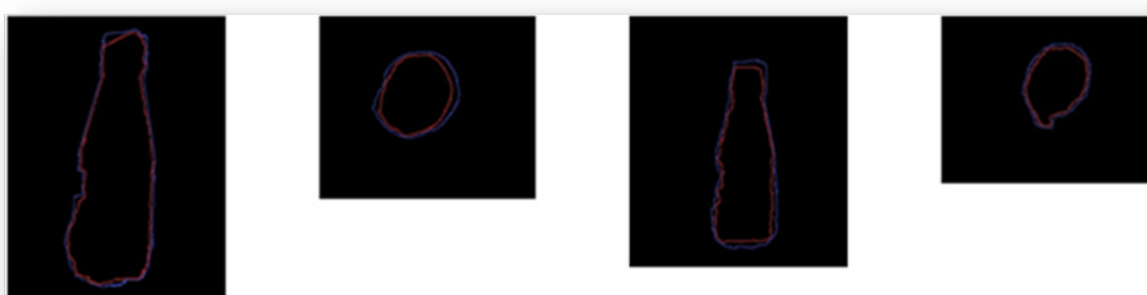


Figure 6. Minimum and maximum feature vector visualization for geometrical model to ketchup bottle.

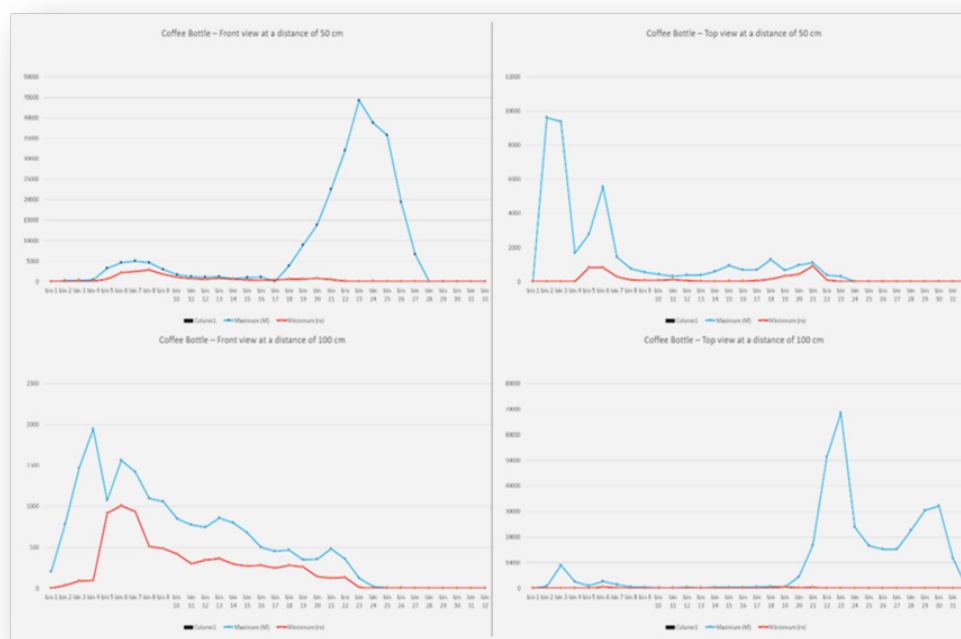


Figure 7. Minimum and maximum feature vector visualization for histogram bin to coffee bottle.

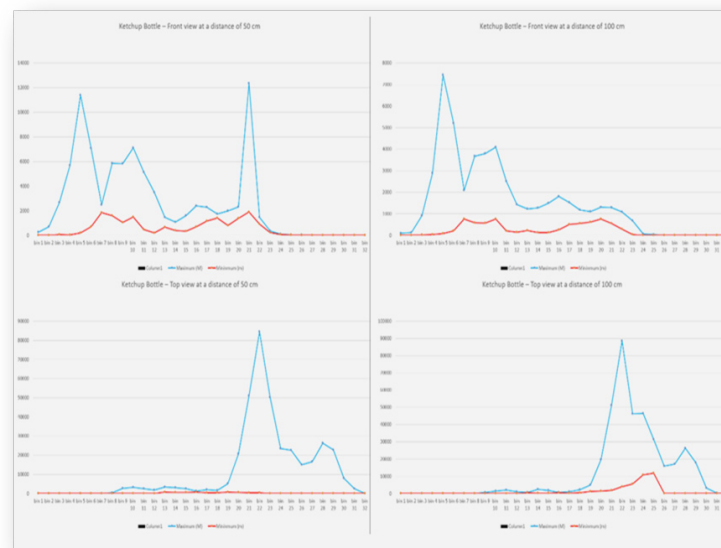


Figure 8. Minimum and maximum feature vector visualization for histogram bin to ketchup bottle.

to primarily lie between the blue and the red line. Else, the test image does not belong to the category.

4. Results and Analysis

We have experimented the algorithm in three different cases, namely, Identification of Coffee bottle, Identification of Ketchup bottle and Object not containing training data set. The first two cases include the test input that was trained. It should intuitively give a positive result. In the third case, we have used a coffee cup, which is not contained in the training data set. The results are conferred in this section.

Case 1: Identification of coffee bottle

The front view of the coffee bottle at the distance of 50 cm is fed as input and is shown in figure 9. The graphical representation of the minimum and maximum feature vector of Geometric model for the training data set in the front view against the Geometric model of the test input is illustrated in figure 10. The largest positive value is displayed as CF50 in the diagonal rank matrix described below, thus, noticeably identifying the target object as coffee bottle, front view at the distance of 50 cm; same as visualized in figure 10a–d.

Diagonal Rank Matrix (The following, is a screenshot of the output.)

```
>>>
CF50      |      24966 0 0 0 0 0 0 0
CF100     |      0 -319861 0 0 0 0 0 0
CT50      |      0 0 -92079 0 0 0 0 0
CT100     |      0 0 0 -103880 0 0 0 0
KF50      |      0 0 0 0 -99120 0 0 0
KF100     |      0 0 0 0 0 -185234 0 0
KT50      |      0 0 0 0 0 0 -431953 0
KT100     |      0 0 0 0 0 0 0 -591592
```

```
CF50      24966
>>> |
```

Where,

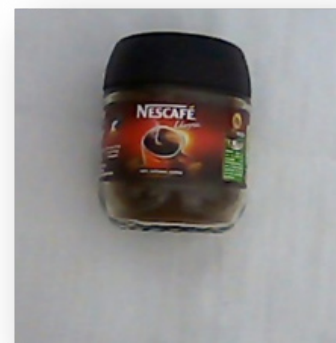


Figure 9. Input Image for coffee bottle.

CF50 - represents coffee bottle front view at distance 50 cm.
CF100- represents coffee bottle front view at distance 100 cm.
CT50 -represents coffee bottle top view at distance 50 cm.
CT100- represents coffee bottle top view at distance 100 cm.
KF50 -represents ketchup bottle front view at distance 50 cm.
KF100- represents ketchup bottle front view at distance 100 cm.
KT50 - represents ketchup bottle top view at distance 50 cm.
KT100- represents ketchup bottle top view at distance 100 cm.

A similar representation for the Histogram bin of the trained data set of the coffee bottle against the test input has been illustrated in figure 11.

Diagonal Rank Matrix

```
>>>
CF50      |      51810752  0  0  0  0  0  0  0  0
CF100     |      0 34961524  0  0  0  0  0  0  0
CT50      |      0  0 -9184267  0  0  0  0  0  0
CT100     |      0  0  0 11079289  0  0  0  0  0
KF50      |      0  0  0  0 5328768  0  0  0  0
KF100     |      0  0  0  0  0 9518040  0  0  0
KT50      |      0  0  0  0  0  0 -61488388  0
KT100     |      0  0  0  0  0  0  0 -1583887809

CF50      51810752
>>> |
```

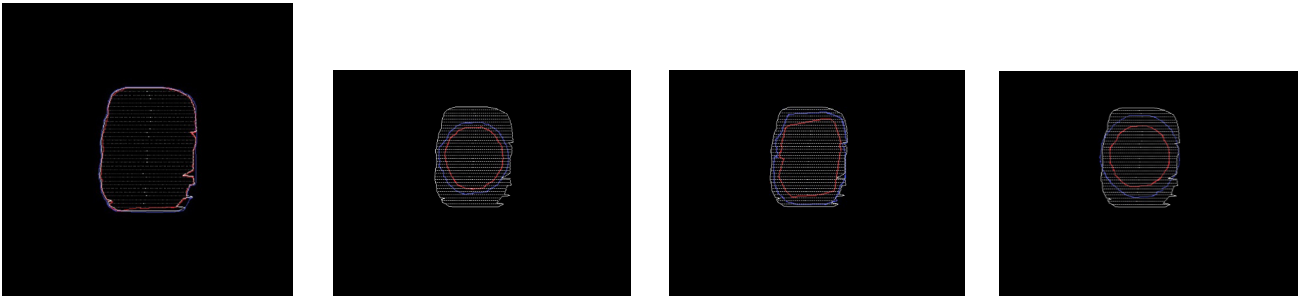


Figure 10 (a-d). Minimum and maximum feature vector visualization for geometrical model of the coffee bottle vs training data set.

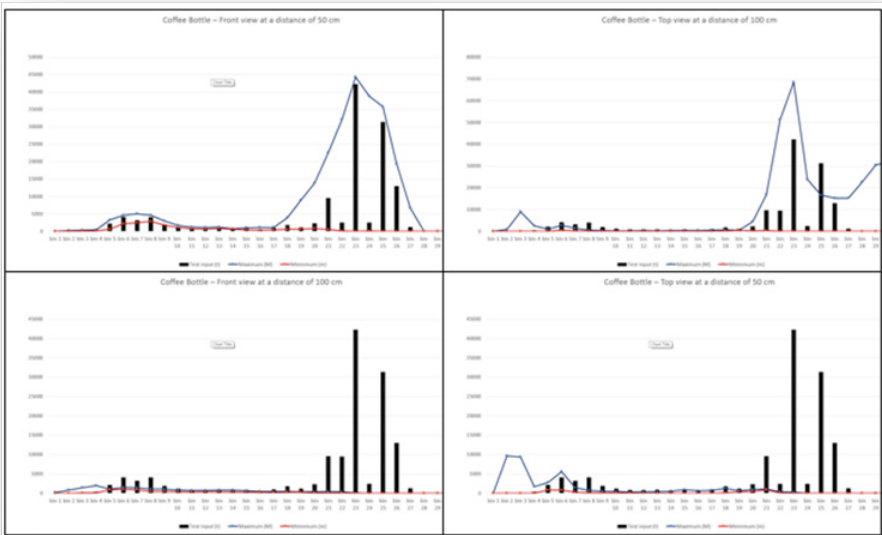


Figure 11. Histogram bin for coffee bottle Vs Training data set.

As it can be observed, for case 1, the outout generated is CF50, that is, coffee bottle identified from the front view at a distance of 50cm, thereby correctly categorising the input.

Case 2: Identification of ketchup bottle

The front view of the ketchup bottle at the distance of 100 cm is taken from training data set. Figure 12 illustrates the same. The graphical representation of the minimum and maximum Feature vector of Geometric model versus training data set is shown in figure 13 (a-d). The largest positive value is represented as KF100 in the diagonal rank matrix described below. Evidently, the target object is identified as ketchup bottle at front view at the distance 100 cm. Same has been visualized in figure 13c., which is ketchup bottle front view at distance 100 cm and training data set front view at distance 100 cm.

Diagonal Rank Matrix

```
>>>
CF50      |      -192979 0 0 0 0 0 0 0
CF100     |      0 -9788 0 0 0 0 0 0
CT50      |      0 0 -162484 0 0 0 0 0
CT100     |      0 0 0 -17698 0 0 0 0
KF50      |      0 0 0 0 -16066 0 0 0
KF100     |      0 0 0 0 0 3013 0 0
KT50      |      0 0 0 0 0 0 -28157 0
KT100     |      0 0 0 0 0 0 0 -18295

KF100     3013
>>> |
```

A similar representation for the histogram bin has been illustrated in figure 14.

The greatest value is identified to the category KF100, as show in the following DRM.

Diagonal Rank Matrix

```
>>>
CF50      |      -42702790 0 0 0 0 0 0 0
CF100     |      0 4920641 0 0 0 0 0 0
CT50      |      0 0 1036340 0 0 0 0 0
CT100     |      0 0 0 5292993 0 0 0 0
KF50      |      0 0 0 0 432602 0 0 0
KF100     |      0 0 0 0 0 13409037 0 0
KT50      |      0 0 0 0 0 0 -16242997 0
KT100     |      0 0 0 0 0 0 0 -1565665503

KF100     13409037
>>> |
```

Case 3: Objectnot containing training data set

The front view of the coffee cup at the distance of 100 cm is given as a test input image, which is not present in the

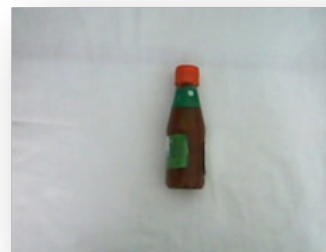


Figure 12. Input image for ketchup bottle.

training data set. Figure 15 illustrates the same. The diagonal rank matrix of the geometrical model and Histogram bin, respectively, presented below reveals that the object is not identified to any of the trained category, since the diagonal matrix has got all non-positive numbers.

Diagonal Rank Matrix

(Histogram bin)

```
>>>
CF50      |      -217233509 0 0 0 0 0 0 0
CF100     |      0 -754820571 0 0 0 0 0 0
CT50      |      0 0 -935463846 0 0 0 0 0
CT100     |      0 0 0 -423031266 0 0 0 0
KF50      |      0 0 0 0 -717226988 0 0 0
KF100     |      0 0 0 0 0 -839055927 0 0
KT50      |      0 0 0 0 0 0 -189661481 0
KT100     |      0 0 0 0 0 0 0 -1440800097
```

No Match Found

```
>>> |
```

Diagonal Rank Matrix

(Geometric modeling)

```
>>>
CF50      |      -310306 0 0 0 0 0 0 0
CF100     |      0 -44145 0 0 0 0 0 0
CT50      |      0 0 -248979 0 0 0 0 0
CT100     |      0 0 0 -45767 0 0 0 0
KF50      |      0 0 0 0 -79725 0 0 0
KF100     |      0 0 0 0 0 -24161 0 0
KT50      |      0 0 0 0 0 0 -71651 0
KT100     |      0 0 0 0 0 0 0 -14346
```

No Match Found

```
>>> |
```

The following table summarizes the statistics of our experimentation using the above methods deploying our proposed algorithm.

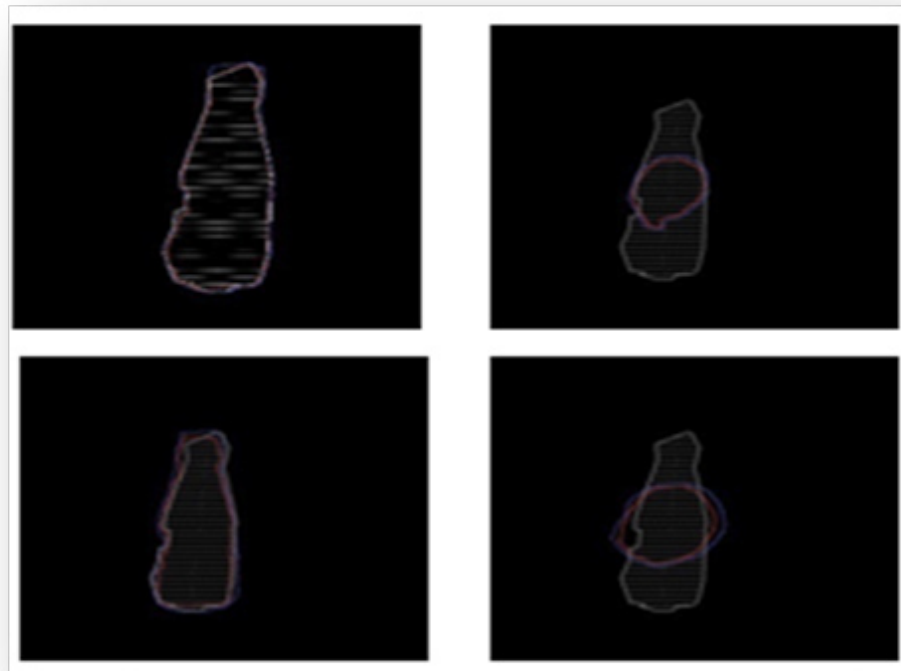


Figure 13 (a-d clockwise). Minimum and maximum feature vector visualization for geometrical model of ketchup bottle vs Training data set.



Figure 14. Histogram bin for ketchup bottle Vs training data set.



Figure 15. Input image for coffee cup.

Table 2. Testing results summary

CATEGORY	Geometric Modeling	Histogram Bin Matching
TOTAL NUMBER OF IMAGES TESTED	35	35
NUMBER OF IMAGES CORRECTLY CLASSIFIED	32	28
NUMBER OF IMAGES INCORRECTLY CLASSIFIED	3	5
NUMBER OF IMAGES NOT IDENTIFIED	0	2
PERCENTAGE OF ACCURACY	91.42 %	80.0 %

5. Conclusion

In this research we have implemented a novel approach called object recognition using feature Weighted Matrix (WM). The proposed method identifies the object in different views namely front and top view. It improves the results in diverse illumination variation conditions for geometrical based feature vector and different scal-

ing variations for both Geometrical based feature vector and Histogram bin feature vector. This feature weighted matrix can be extended to solve multiple objects present in the image and different invariant conditions.

6. References

1. Prasad, Dilip K. Survey of the problem of object detection in real images. *IJIP*. 2012; 6(6): 441.
2. Mundy, Joseph L. Object recognition in the geometric era: A retrospective. Toward category- level object recognition. Berlin Heidelberg: Springer; 2006. p. 3–28.
3. Alexander A, Tsotsos JK. 50 Years of object recognition: directions forward. *Computer Vision and Image Understanding*. 2013; 117(8):827–91.
4. Lee WT, Chen HT. Histogram-based interest point detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2009. p. 1590–6.
5. Zhang H, Gao W, Chen X, Zhao D. Object detection using spatial histogram features. *Image and Vision Computing*. 2006; 24:327–41.
6. Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*. 2007; 3:177–280.
7. Prasad DK, Leung MKH, Quek C. ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method. *Pattern Recogn*; 2013.
8. Prasad DK, Leung MKH. Error analysis of geometric ellipse detection methods due to quantization. In *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT 2010)*. Singapore; 2010. p. 58–63.
9. Olson CF. A general method for geometric feature matching and model extraction. *Int J Comput Vis*. 2001; 45:39–54.
10. Choi O, Kweon IS. Robust feature point matching by preserving local geometric consistency. *Comput Vis Image Understand*. 2009; 113:726–42.
11. Ranga A, Suryanarayana G. A Novel approach for single image super resolution by sparse signal representation. *Indian Journal of Science and Technology*. 2015 Jan; 8(S2):74–7.