Classification of Kitchen Cutlery using a Visual Recognition Algorithm

Arushi Goel
IIIT Allahabad
iec2013078@iiita.ac.in

Robert B. Fisher
University of Edinburgh
rbf@inf.ed.ac.uk

Abstract

The classification of kitchen cutlery has a variety of applications ranging from assisting physically disabled individuals to helping in daily household chores. A hierarchical classifier with feature selection is presented for kitchen cutlery classification. It recognizes 20 classes of cutlery with a total of 897 images that vary from class to class. Given the amount of noise and shape variations present in the segmented images and features, it was challenging to achieve a high accuracy rate for the given set of classes. Finally, an average accuracy of 86% was achieved with some improvements.

1. Introduction

With the growth and advent of technology in the 21st century, every piece of human work is getting automated. This kind of automation has reduced human effort and enhanced their focus on more difficult and challenging jobs instead of the day to day menial jobs. The way technology and robotics has entered our homes is astonishing. This trend is growing exponentially and will continue to grow with young researchers and engineers pioneering their ideas in this field. From automatic lights, fans, heaters and other electronic devices, imagine a robot helping you in the kitchen to cook a delicious meal whilst you sit leisurely. This robot could also be helpful in washing the dishes, lending you a helping hand while you are cooking, cleaning the table after the meal is done and also preparing the whole meal.

Some previous work for classification of kitchen cutlery has been done. D.Fullerton [1] achieved an average recognition rate of 69% with 18 classes of kitchen cutlery. The idea behind our work is to improve the effectiveness of the classifier, increase the database and extend the website. The main contributions of this project are – (1) Increase the database from the existing 449 images [1]. (2) Include more features to obtain more information from the images. (3) A Hierarchical classifier to take care of the greatly confused set of classes. (4) Extend the website to include the new set of images.

2. Dataset

2.1 Data Collection

The original database of 449 images [1] was extended by adding 448 more images thereby increasing the total to 897 images. The images were collected by visiting charity shops and bargain stores in Edinburgh. These images had to be manually captured to ensure high resolution, proper lighting conditions and to maintain a constant background for the images. The choice of the green background [1] seemed to be of valuable use as the object stood out in the image with that background. As most utensils are of silver texture it was easy to segment the images using chroma keying.

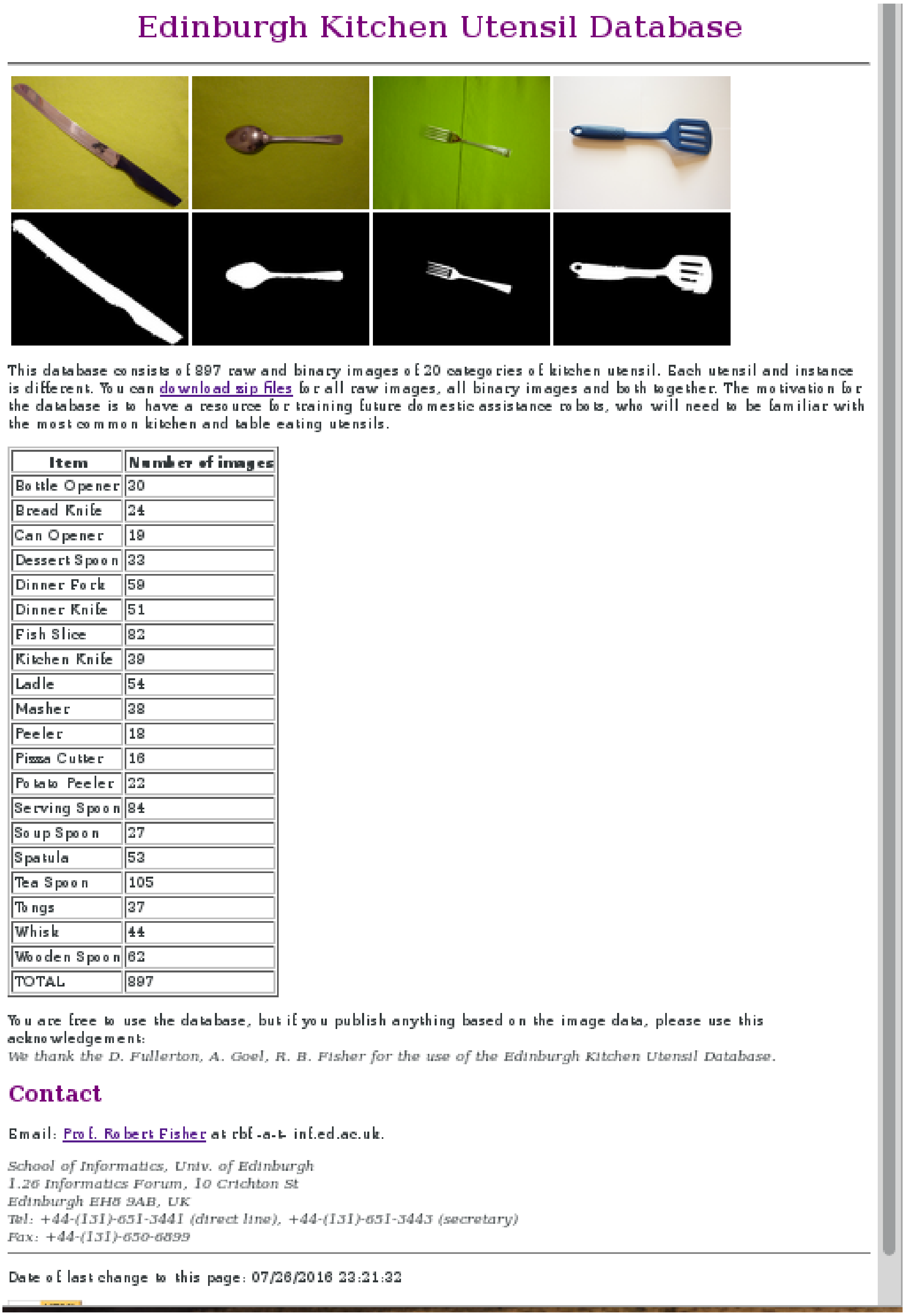
The task of manually capturing the images was time consuming and involved visiting many shops for different designs of utensils, but it had to be done with interest for proper results. Most of the utensils in the shops were either wrapped or tagged and it was hence difficult to include them in the database. This led to fewer images in some classes. The exact number of images in each class including the images from [1] are shown in table 2.1.

ITEM	Number of Images
Bottle Opener	30
Bread Knife	24
Can Opener	19
Dessert Spoon	33
Dinner Fork	59
Dinner Knife	51
Fish Slice	82
Kitchen Knife	39
Ladle	54
Masher	38
Peeler	18
Pizza Cutter	16
Potato Peeler	22
Serving Spoon	84
Soup Spoon	27
Spatula	53
Tea Spoon	105
Tongs	37
Whisk	44
Wooden Spoon	62
TOTAL	897

Table 2.1

2.2 Website Creation

The website created [1] was extended to include the new database with download links to all the raw and binary images for each class (URL - http://homepages.inf.ed.ac.uk/rbf/UTENSILS/). It uses xml which provides an easy way to present the image data. Each class has its own separate xml file that displays all raw and binary images in that particular class. The website is shown in figure 2.1 and 2.2.



Edinburgh Kitchen Utensil Database

This database consists of 897 raw and binary images of 20 categories of kitchen utensil. Each utensil and instance is different. You can download zip files for all raw images, all binary images and both together. The motivation for the database is to have a resource for training future domestic assistance robots, who will need to be familiar with the most common kitchen and table eating utensils.

Item	Number of images
Bottle Opener	30
Bread Knife	24
Can Opener	19
Dessert Spoon	33
Dinner Fork	59
Dinner Knife	51
Fish Slice	82
Kitchen Knife	39
Ladle	54
Masher	38
Peeler	18
Pizza Cutter	16
Potato Peeler	22
Serving Spoon	84
Soup Spoon	27
Spatula	53
Tea Spoon	105
Tongs	37
Whisk	44
Wooden Spoon	62
TOTAL	897

You are free to use the database, but if you publish anything based on the image data, please use this acknowledgement:
We thank the D. Fullerton, A. Goel, R. B. Fisher for the use of the Edinburgh Kitchen Utensil Database.

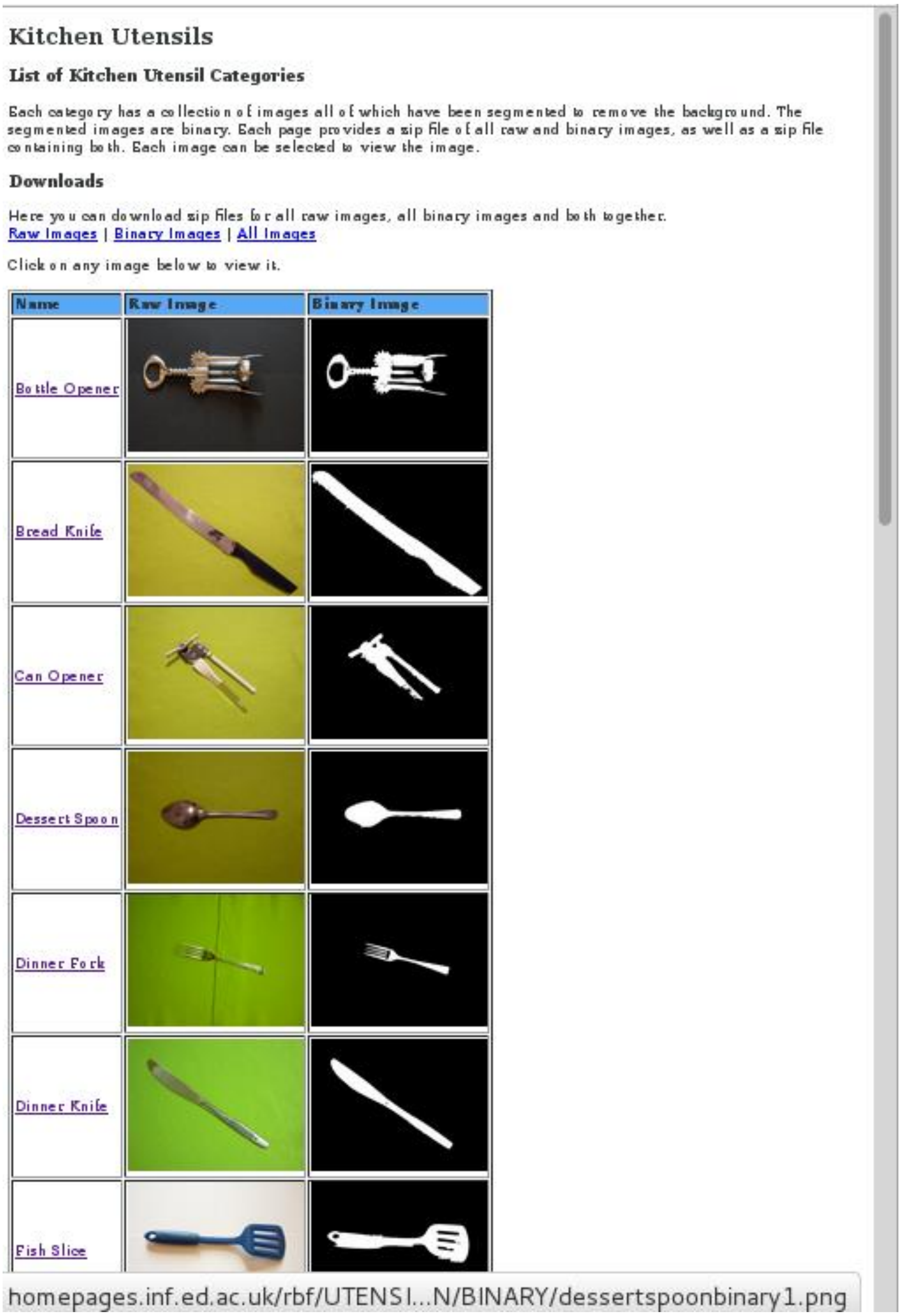
Contact

Email: Prof. Robert Fisher at rbf -a-t- inf.ed.ac.uk.

School of Informatics, Univ. of Edinburgh
1.26 Informatics Forum, 10 Crichton St
Edinburgh EH8 9AB, UK
Tel: +44-(131)-651-3441 (direct line), +44-(131)-651-3443 (secretary)
Fax: +44-(131)-650-6899

Date of last change to this page: 07/26/2016 23:21:32

Fig 2.1



Kitchen Utensils

List of Kitchen Utensil Categories

Each category has a collection of images all of which have been segmented to remove the background. The segmented images are binary. Each page provides a zip file of all raw and binary images, as well as a zip file containing both. Each image can be selected to view the image.

Downloads

Here you can download zip files for all raw images, all binary images and both together.
Raw Images | Binary Images | All Images

Click on any image below to view it.

Name	Raw Image	Binary Image
Bottle Opener		
Bread Knife		
Can Opener		
Dessert Spoon		
Dinner Fork		
Dinner Knife		
Fish Slice		

homepages.inf.ed.ac.uk/rbf/UTENSI...N/BINARY/dessertspoonbinary1.png

Fig 2.2

3. Methodology

In this section, we present the steps that were used to classify the kitchen cutlery. The raw images obtained have to be thresholded first to obtain the binary images. Then features are extracted from these binary images to train the classifier. A hierarchical classifier using SVM and Bayes is described that improves the accuracy at the higher levels of hierarchy.

3.1 Thresholding

The three channels in the RGB image are first normalized to scale the pixel values to [0,1] from [0,255]. The red, green and blue channels in the normalized RGB image are checked for their values. The channel which has a different range of values for the object and the background is then used to segment the image into binary. Each image has to be individually checked for these values and then the algorithm has to be improved every time for each image to get a binary image. The distinct background of green used for taking the images helped in removing any ambiguity that we would have faced if we used a white or black background.

After this step, a median filter of window size 10 (generally but adapted frequently) is applied to the binary image to smooth the edges and remove any spurs from the background. This is an important process as it ensures that the final image that will be used for feature extraction is free from noise. A raw image and its binary equivalent are shown in figure 3.1 and 3.2.

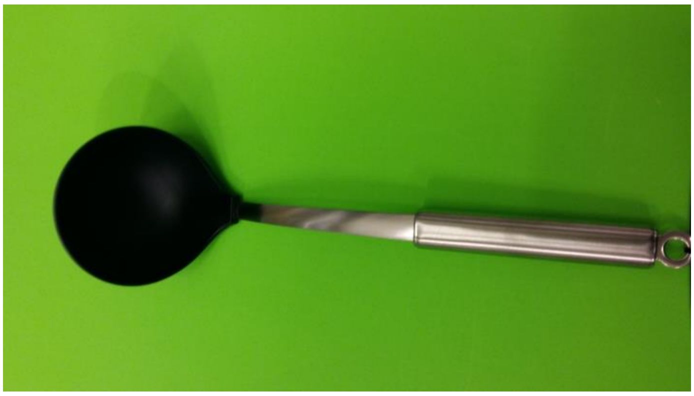


Fig 3.1 Raw Image

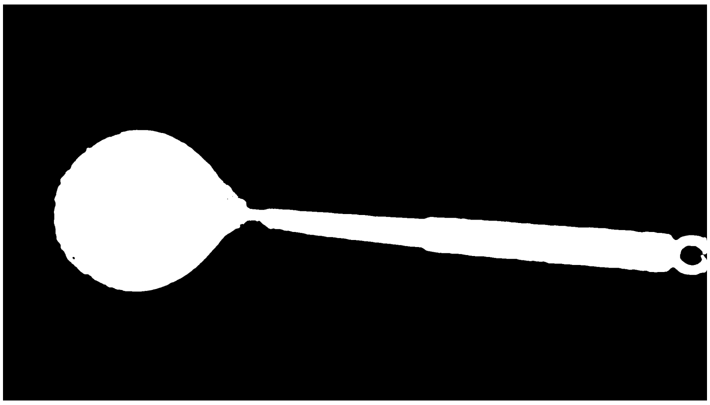


Fig 3.2 Binary Image

3.2 Feature Extraction

Seventeen features (8 old and 9 new features) were calculated for each image in the dataset and every feature was invariant to scaling, rotation and translation. Invariance is important because the image that will be captured by the robot can be from any distance, direction and axis. These differences in the images should not affect their feature values and hence their recognition. Compactness [1], six moment invariants [2] and prongs [1] were also used. The new features are listed below -

- **Convex** – This feature describes the convex hull property. The ratio of the area of the shape to the area of the convex hull is calculated. For utensils like knives, this value will be close to 1, whereas, for spoons and forks this value will be around 0.7 or 0.8.

- **Skeleton** – This feature uses the skeletonization property of binary images. It is useful for distinguishing between uniform images and images with more variations. The skeletons of images like spoons, knives and tongs are more uniform than bottle openers, whisks , mashers and forks. Although some arbitrary values came into picture due to the roughness of the edges and hence the skeletonized image was smoothed for appropriate final values. The ratio of the perimeter of the original image to that of the skeletonized image is used as the feature values.

- **Elongation** – This property proved to be useful for classes that can be distinguished based on their symmetry about the centre of mass or elongation axis. The image was divided into two parts about the centre of mass and perpendicular to the elongation axis and then areas of the two parts was calculated. The ratio of the area of smaller part to the area of the larger part was noted. For classes like bottle openers, can openers, dinner knives, bread knives, kitchen knives, this value is close to 1 because they are almost symmetric about the centre of mass, whereas for tea spoons, dessert spoons, wooden spoons, whisks, the value should be less than 1 because of a thin handle at one end.

- **Erode** – This property is used to separate the classes that have lumps on end end (spoons of all types) from the remaining classes. The image is eroded N times until the whole image disappears. Then the original image is eroded N/2 times and then dilated N/2 times which leaves the lump portion in the dilated image. The ratio of the area of the dilated image to the area of the complete image is then calculated.

- **Hole** – This property classifies the classes that contain holes (significant holes and not the noise in the images) from the remaining set. The image is inverted and the ‘AND’ operation of the filled image with the inverted image gives the holes in the image. The ratio of the area of the holes to the area of the image is then calculated. For utensils like fish slice, whisks and serving spoons which have a significant amount of holes this value is greater than for the classes such as spoons, forks, bottle and can openers, tongs etc.

- **Shape** – For utensils like the spatula and fish slice which have a more rectangular shape at the head than serving spoons and wooden spoons, the head part is isloated by repeated erosions (keeping the part with greatest area in the eroded image) and then a bounding box is drawn around this region. For circular shapes, ratio of the area of the region to the area of the bounding box is less than 1 whereas for rectangular shapes it will be close to 1.

- **Edge** – For classifying between whisk and mashers, the property that masher contains a big hole in between is used. The image is first dilated by a structuring element to remove any small holes and spaces, then the ratio of the area of the dilated image to the area of the original image is calculated. This helped in distinguishing between these two classes.

- **Colour angle** – This property takes into account the colour of the original rgb image. Until now only binary images were used for feature extraction. In this property, the original image is masked with the binary image which removes the background and retains only the object of interest. The mean of the red, green and blue channels is then calculated and formed into a vector. The value ‘colour angle’ is the dot product of this vector with the vector for white colour ([255 255 255] or [1/3,1/3,1/3] (if normalized)). In this way, utensils made of wood or with distinct colours are separated from the ones made of metal.

- **Symmetry** – For classifying between classes such as spatula and wooden spoon, which have different symmetries regarding their heads, this property is applied. The head part of the image is retained by erosions and then we calculate the second moments of that part of the image. The second moment values are different for these two classes.

3.3 Normalization of Features

After obtaining the feature vectors for all classes, these feature values are normalized. This type of normalization reduces the ambiguity in features and the classifier performs better when trained with these normalized features. The normalization used was z-normalization in which a new feature value y is calculated by the following rule -

$$y = (x\text{-}smean)/sstd$$

where x is the original feature value, smean is the mean value of that particular feature taken over all classes and sstd is the standard deviation of that feature taken over all classes.

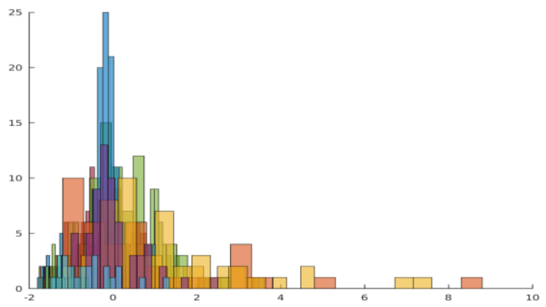
This reduces each value of feature for every image to a scaled value which has mean value 0 and standard deviation of 1.

3.4 Histogram and Gaussian Plots

The normalized features values for each class are then used to plot histograms. Firstly, histogram and Gaussian plots for each class and each feature are plotted separately. Then with all the classes taken together, a combined histogram and Gaussian plot is drawn for each feature separately. This helps in analysing the feature values in two ways before actually sending them to the classifier-

- **Outliers** – The individual plots show that the image which has a different value of the feature in each class is an outlier and its value can affect the training if the value is more spread out than the rest of the data.
- **Separability** - This is more noticeable in the combined Gaussian plots of all classes. If most of the classes have feature values around the same area on a particular axis, it means that that description will not be very effective in differentiating between the two classes. If this is the case for many classes then it may be worth removing that description entirely. On the other hand if the values do not overlap that much in a particular axis then this description has high separability and will help to recognise the class of a utensil or atleast help in separating groups of classes.

These plots thus provide a good way for checking the effectiveness of the feature values for all classes. Some histogram plots for the features are shown in figures 3.3 – 3.17.



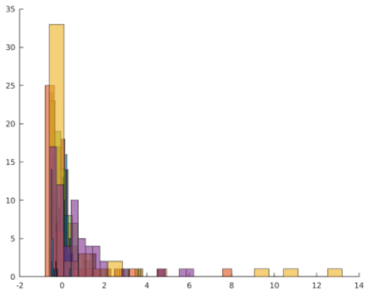
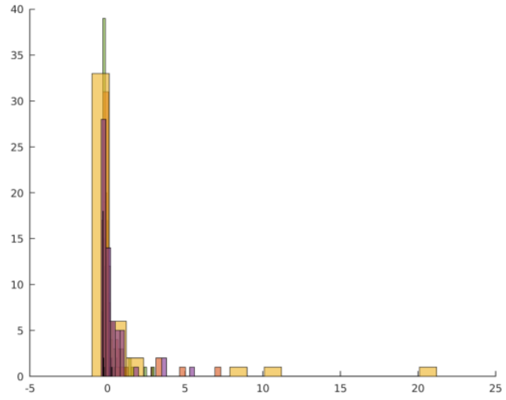


Fig 3.4 ci2



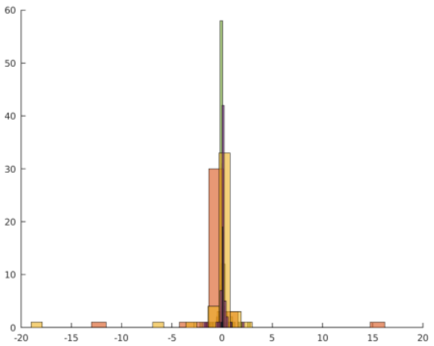


Fig 3.5 ci3

Fig 3.6 ci4

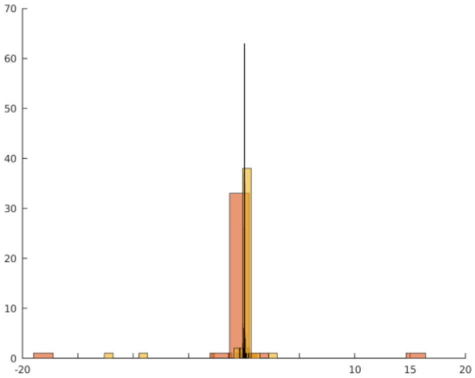


Fig 3.7 ci6

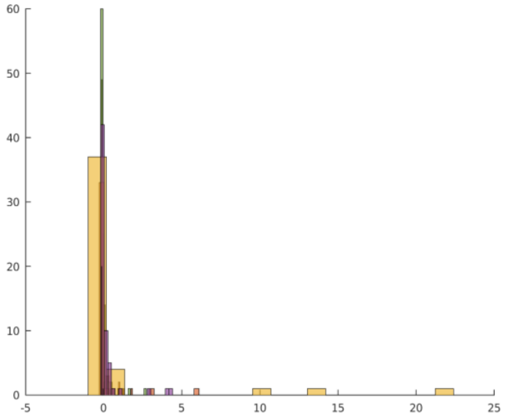


Fig 3.8 ci5

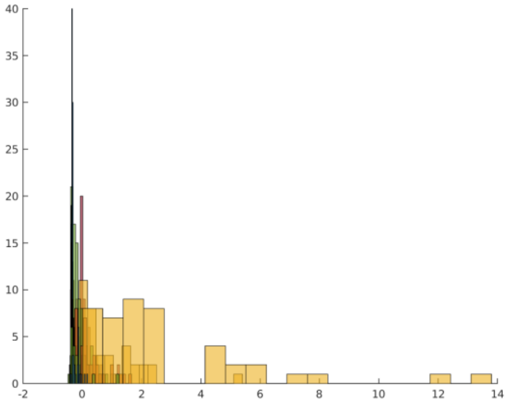


Fig 3.9 Compactness

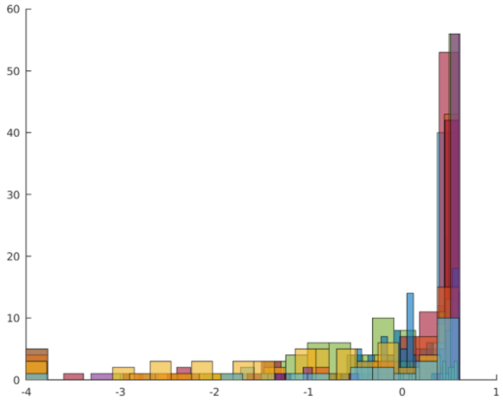


Fig 3.10 Prongs

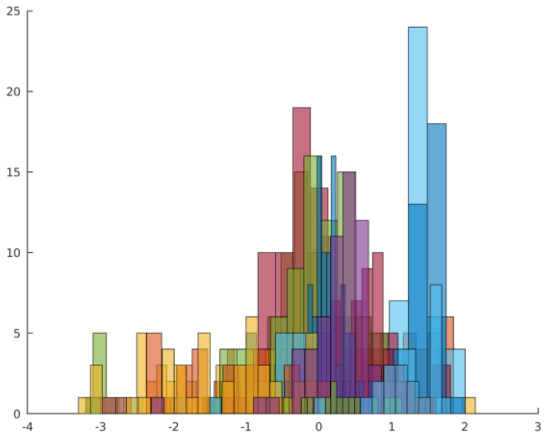


Fig 3.11 Convex

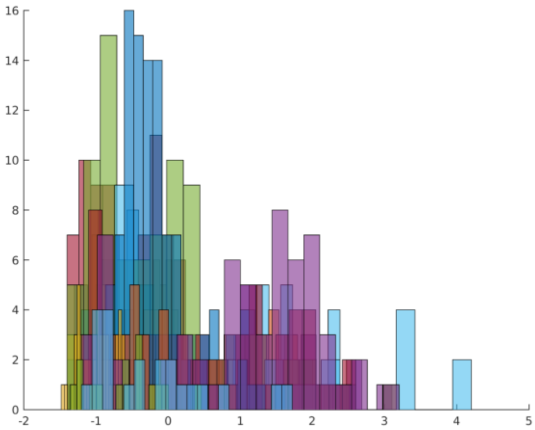


Fig 3.12 Elongation

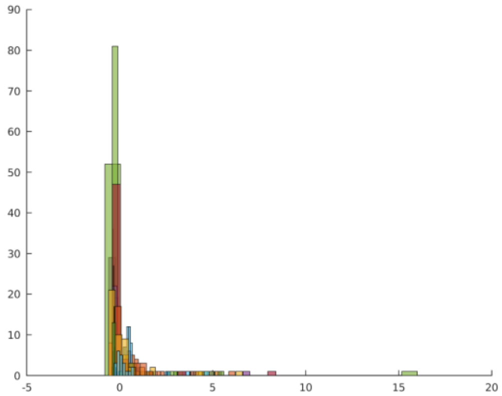


Fig 3.13 Symmetry

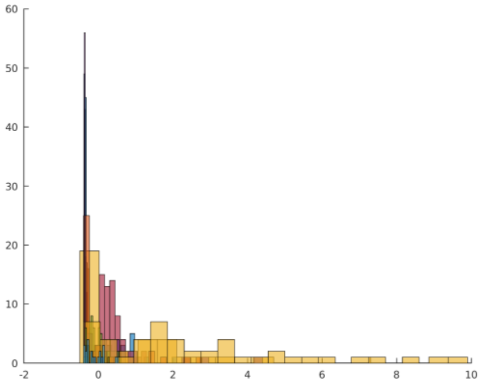


Fig 3.14 Hole

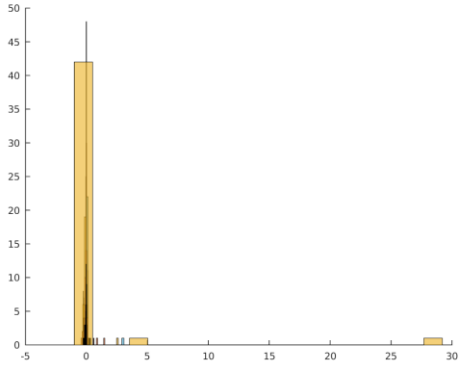


Fig 3.15 Shape

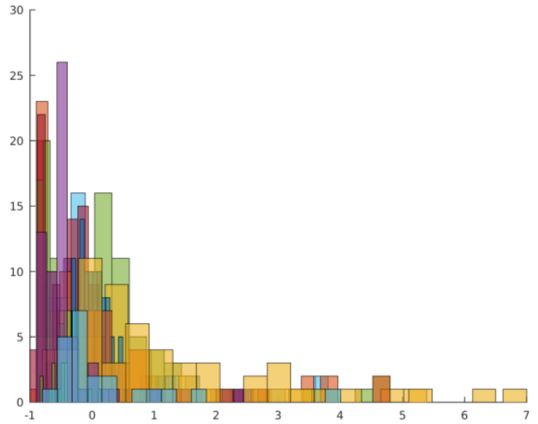


Fig 3.16 Edge

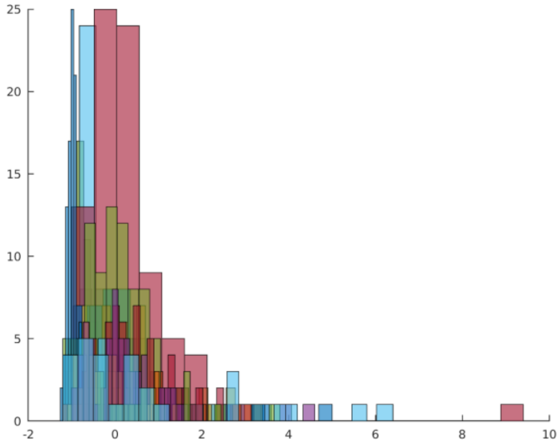


Fig 3.17 Colour Angle

3.5 Recognition

The accuracy obtained using Bayes Classifier [1] was approximately 69% with the sixth moment invariant feature removed (i.e. ci6). This accuracy was calculated over 18 classes due to fewer images in some of the classes. The introduction of 9 new features and increase in the database was helpful in classifying all the 20 classes together with an increase in the accuracy. The following improvements were made to the Bayes classifier [1] – (1) Normalization of Feature values used to train the classifier. (2) Forward Sequential feature selection to choose the subset of features that produce the maximum accuracy.

Normalization of feature values was done as described in section 3.3. The algorithm for forward sequential feature selection is described below -

- Start with an empty feature set, say **f.**
- Begin with the first feature and going over all features, compute the accuracy of the classifier. Pick up that feature which yields the highest accuracy and add it to the set f.
- Remove this feature added to f from the original feature set and then start the same process again over the remaining features till the original feature set becomes empty.
- Then check which feature subset in **f** produced the maximum accuracy. This will give the final set of features.

3.6 Hierarchical Classifier

We present a hierarchical classifier to divide the classes in the form of a binary tree so as to improve more similar classes at higher levels of hierarchy with some customized classifiers. The classes are divided at each level such that the accuracy for the left versus right group of classes is the maximum. One way is to try all such combinations that produce the maximum accuracy and the other way is to check one vs the remaining class accuracy and the classes that have accuracy above some threshold can be chosen to be on one side and the rest on the other. It is important to have higher accuracy at the top level because if an image from the right class went to the left class then it will end up getting classified as wrong.

For each level of hierarchy either an SVM or a Bayes classifier can be used for classification whichever produces the maximum accuracy for those set of classes.

A two-class SVM classifier is considered good for separation of two classes as it is a maximum margin classifier. Hence it can be used to divide the classes into the left and the right classes of a hierarchical tree. At high levels of hierarchy when there is no further division, either a Bayes or a multiclass SVM classifier can be used for classification of the given image. A classifier is chosen depending on the accuracy given by each.

The tree created for classification in this case is shown in figure 3.20 with the class codes as given in table 3.1.

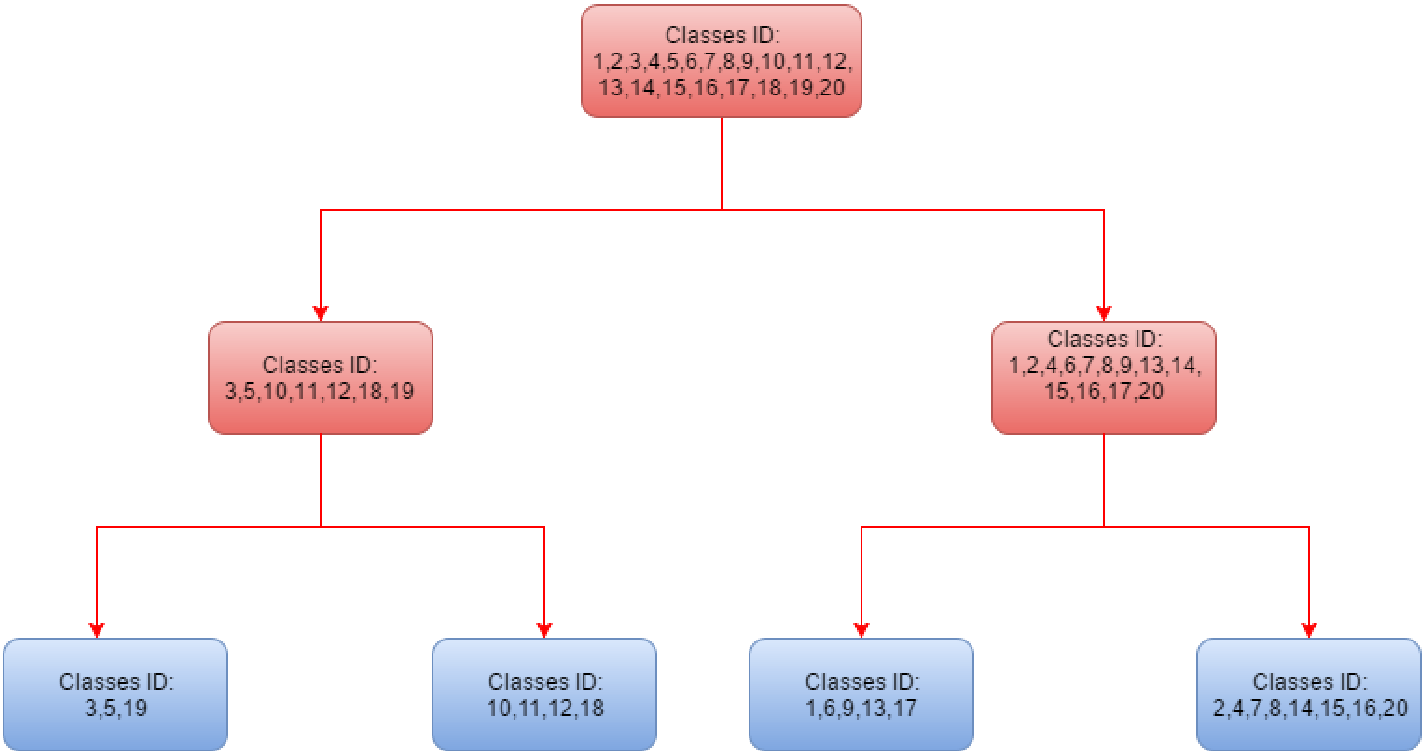


Fig 3.1 Hierarchy Tree

Color Pink –Two class SVM classifier for division into left and right side of classes

Color Blue – Bayes Classifier

CLASSES	**ID**
BOTTLE OPENER	1
BREAD KNIFE	2
CAN OPENER	3
DESSERT SPOON	4
DINNER FORK	5
DINNER KNIFE	6
FISH SLICE	7
KITCHEN KNIFE	8

LADLE	9
MASHER	10
PEELER	11
PIZZA CUTTER	12
POTATO PEELER	13
SERVING SPOON	14
SOUP SPOON	15
SPATULA	16
TEA SPOON	17
TONGS	18
WHISK	19
WOODEN SPOON	20

Table 3.1 Class Codes

4. Implementation

The Bayes classifier described in section 3.5 was chosen as a baseline classifier and we recognize that it was optimistic because no independent test data was used. The training and testing stages were implemented using all the set of images. A confusion matrix is created where the row of the matrix represents the class of the image that is predicted by the Bayes classifier and the column represents the class the image actually represents. There are two ways to measure the performance of the classifier using the confusion matrix – (1) Macro Accuracy – It computes the average correct classification over all images. (2) Micro Accuracy – It computes the accuracy for each individual class and then takes the average of the accuracies.

The confusion matrix obtained from the above Bayes classifier for the case of maximum accuracy using feature selection is used at the level 0 of hierarchy to separate the classes into left and right. The technique used to decide which class should go to which side depended on maximizing the accuracy of left classes versus right classes. The method of one vs remaining classes was used, where one class at a time was considered as being on the left side and the remaining on the right side. Then the confusion accuracy of the left vs right classes was obtained. In this way we got 20 confusion accuracies considering each class one at a time. Then, those classes with accuracy above 98.5% were kept on the left side and the remaining on the right side. Similar process was followed for further division at level 1 for both the left and right side of classes.

While testing the accuracy of the hierarchical classifier, each image at level 0 is decided to be in the left or right class using a pre-trained two-class SVM classifier. Then, again at level 1 SVM is used to further divide the classes. At level 2, when two child nodes for each side of level1 are obtained, a Bayes Classifier with feature selection is used to further classify the images in that particular group. A multi class SVM classifier can also be used for classifying the group of classes at the highest level of hierarchy. But the accuracy obtained with Bayes classifier was better than that with multi class SVM.

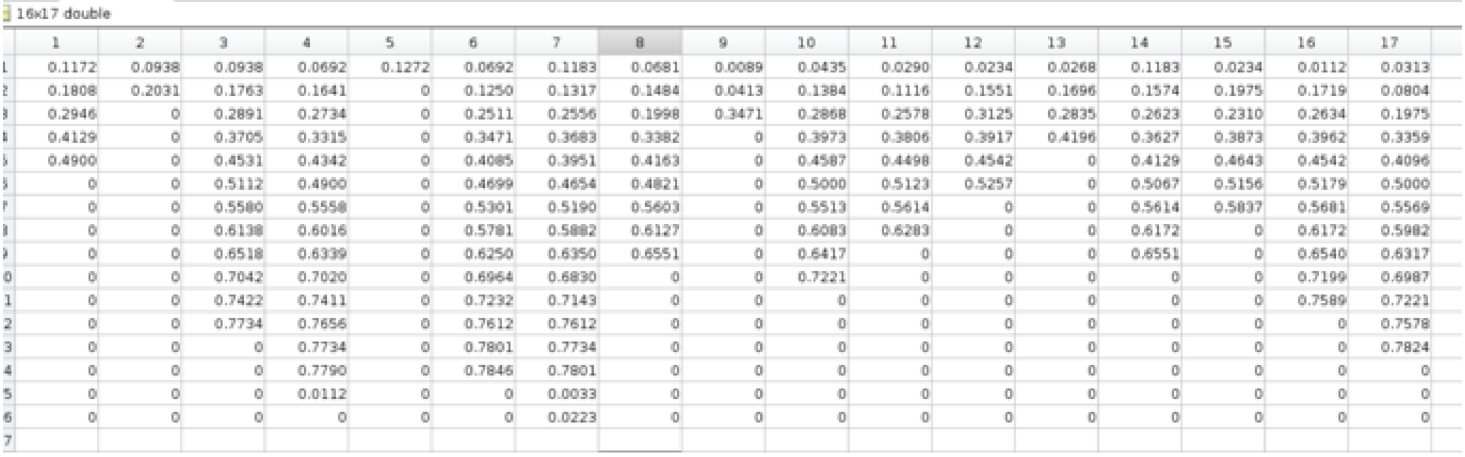
This leads to each image being classified in at-least one of the classes and helps us to use more customized set of features at higher levels of hierarchy for the confused classes.

5. Results

5.1 Bayes Classifier

Applying the Bayes classifier to all the 20 classes along with feature selection where the features are in the order - 'Compactness','ci1','ci2','ci3','ci4','ci5','ci6','Prongs','Convex','Skeleton','Elongation','Erosion','Hole','Shape','Edge','Angle','Symmetry' ,the accuracies are obtained as shown in the figure 6.1. According to this matrix, we can see that the maximum accuracy is for the row 14 i.e. **78.46%**. So removing the features ‘ci3’ and ‘ci6’, the macro accuracy is **81.41%** and the micro accuracy is **78.46%**. The confusion matrix using these set of features is shown in figure 6.2.

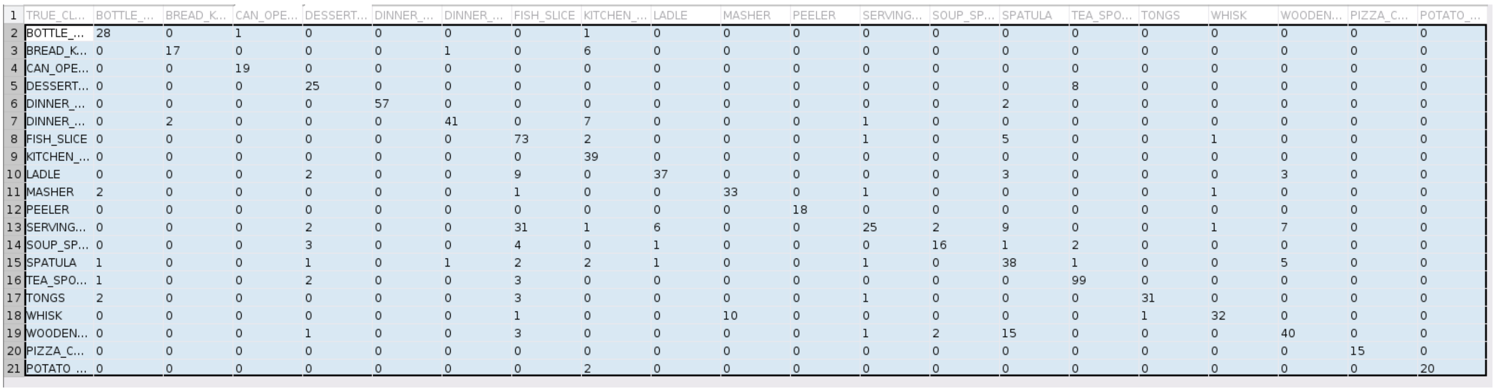
Each row in the matrix shown in Fig 6.1 corresponds to the accuracy that we obtained after selecting a new feature in each row and the features that produce the maximum accuracy in the previous rows. For example, in row 1 the maximum accuracy is 0.1272(column 5) i.e. we select the feature ci4 as it has produced this maximum accuracy. Then in row 2 keeping the feature ci4 from above and then trying ci4 with every other feature remaining, we again check the maximum accuracy in row 2. We see that 0.2031 is maximum in row 2 and this corresponds to the feature ci1, so now two features ci4 and ci1 are selected. This procedure goes until all the features are tested and then the matrix is checked for the column with maximum accuracy.



16x17 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.1172	0.0938	0.0938	0.0692	0.1272	0.0692	0.1183	0.0681	0.0089	0.0435	0.0290	0.0234	0.0268	0.1183	0.0234	0.0112	0.0313
2	0.1808	0.2031	0.1763	0.1641	0	0.1250	0.1317	0.1484	0.0413	0.1384	0.1116	0.1551	0.1696	0.1574	0.1975	0.1719	0.0804
3	0.2946	0	0.2891	0.2734	0	0.2511	0.2556	0.1998	0.3471	0.2868	0.2578	0.3125	0.2835	0.2623	0.2310	0.2634	0.1975
4	0.4129	0	0.3705	0.3315	0	0.3471	0.3683	0.3382	0	0.3973	0.3806	0.3917	0.4196	0.3627	0.3873	0.3962	0.3359
5	0.4900	0	0.4531	0.4342	0	0.4085	0.3951	0.4163	0	0.4587	0.4498	0.4542	0	0.4129	0.4643	0.4542	0.4096
6	0	0	0.5112	0.4900	0	0.4699	0.4654	0.4821	0	0.5000	0.5123	0.5257	0	0.5067	0.5156	0.5179	0.5000
7	0	0	0.5580	0.5558	0	0.5301	0.5190	0.5603	0	0.5513	0.5614	0	0	0.5614	0.5837	0.5681	0.5569
8	0	0	0.6138	0.6016	0	0.5781	0.5882	0.6127	0	0.6083	0.6283	0	0	0.6172	0	0.6172	0.5982
9	0	0	0.6518	0.6339	0	0.6250	0.6350	0.6551	0	0.6417	0	0	0	0.6551	0	0.6540	0.6317
10	0	0	0.7042	0.7020	0	0.6964	0.6830	0	0	0.7221	0	0	0	0	0	0.7199	0.6987
11	0	0	0.7422	0.7411	0	0.7232	0.7143	0	0	0	0	0	0	0	0	0.7589	0.7221
12	0	0	0.7734	0.7656	0	0.7612	0.7612	0	0	0	0	0	0	0	0	0	0.7578
13	0	0	0	0.7734	0	0.7801	0.7734	0	0	0	0	0	0	0	0	0	0.7824
14	0	0	0	0.7790	0	0.7846	0.7801	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0.0112	0	0	0.0033	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0.0223	0	0	0	0	0	0	0	0	0	0
17																	

Fig 5.1 Feature Selection Accuracy Matrix



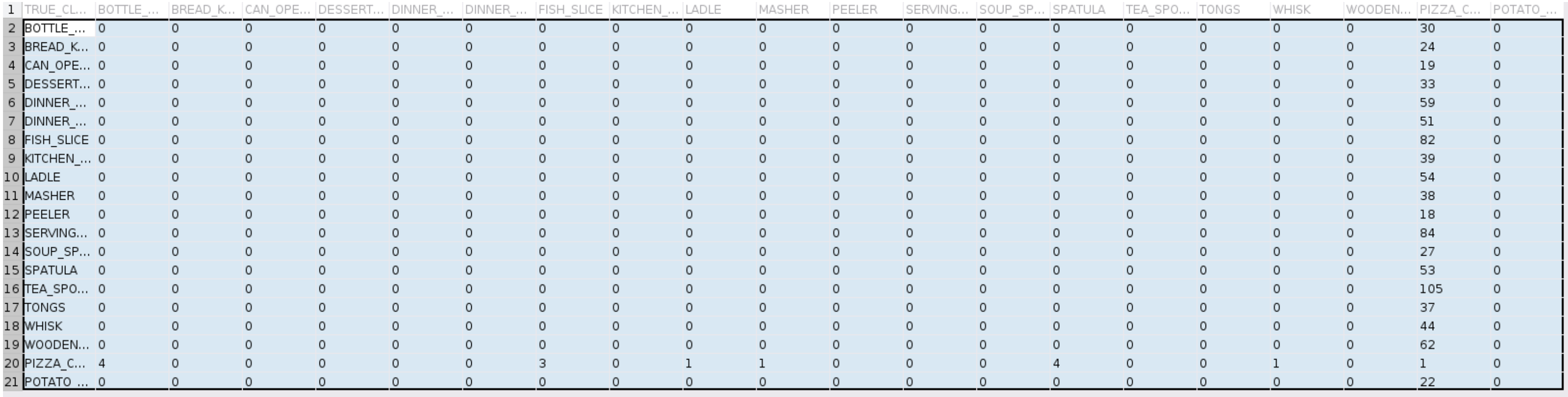
1	TRUE_CL...	BOTTLE_...	BREAD_K...	CAN_OPE...	DESSERT...	DINNER_...	DINNER_...	FISH_SLICE	KITCHEN_...	LADLE	MASHER	PEELER	SERVING...	SOUP_SP...	SPATULA	TEA_SPO...	TONGS	WHISK	WOODEN...	PIZZA_C...	POTATO_...
2	BOTTLE_...	28	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	BREAD_K...	0	17	0	0	0	1	0	6	0	0	0	0	0	0	0	0	0	0	0	0
4	CAN_OPE...	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	DESSERT...	0	0	0	25	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0
6	DINNER_...	0	0	0	0	57	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
7	DINNER_...	0	2	0	0	0	41	0	7	0	0	0	1	0	0	0	0	0	0	0	0
8	FISH_SLICE	0	0	0	0	0	0	73	2	0	0	0	1	0	5	0	0	1	0	0	0
9	KITCHEN_...	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0
10	LADLE	0	0	0	2	0	0	9	0	37	0	0	0	0	3	0	0	0	3	0	0
11	MASHER	2	0	0	0	0	0	1	0	0	33	0	1	0	0	0	0	1	0	0	0
12	PEELER	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0
13	SERVING...	0	0	0	2	0	0	31	1	6	0	0	25	2	9	0	0	1	7	0	0
14	SOUP_SP...	0	0	0	3	0	0	4	0	1	0	0	0	16	1	2	0	0	0	0	0
15	SPATULA	1	0	0	1	0	1	2	2	1	0	0	1	0	38	1	0	0	5	0	0
16	TEA_SPO...	1	0	0	2	0	0	3	0	0	0	0	0	0	0	99	0	0	0	0	0
17	TONGS	2	0	0	0	0	0	3	0	0	0	0	1	0	0	0	31	0	0	0	0
18	WHISK	0	0	0	0	0	0	1	0	0	10	0	0	0	0	0	1	32	0	0	0
19	WOODEN...	0	0	0	1	0	0	3	0	0	0	0	1	2	15	0	0	0	40	0	0
20	PIZZA_C...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
21	POTATO ...	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	20

Fig 5.2 Confusion Matrix for Bayes Classifier with Feature Selection

5.1.1 Discussion

As it can be observed in the matrix shown in Fig 5.1 that the accuracy has unexpectedly fallen steeply after row 14. The confusion matrix obtained after using the set of features in row 15 is shown in figure 5.3.

The classes behave really badly using these set of features which seems like a code bug but there was no more time in the project to investigate this issue. This shows that feature selection helps in finding out the best subset of features for the classifier.



1	TRUE_CL...	BOTTLE_...	BREAD_K...	CAN_OPE...	DESSERT...	DINNER_...	DINNER_...	FISH_SLICE	KITCHEN_...	LADLE	MASHER	PEELER	SERVING...	SOUP_SP...	SPATULA	TEA_SPO...	TONGS	WHISK	WOODEN...	PIZZA_C...	POTATO_...
2	BOTTLE_...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0
3	BREAD_K...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0
4	CAN_OPE...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0
5	DESSERT...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0
6	DINNER_...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0
7	DINNER_...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0
8	FISH_SLICE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	0
9	KITCHEN_...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0
10	LADLE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	0
11	MASHER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
12	PEELER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0
13	SERVING...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0
14	SOUP_SP...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0
15	SPATULA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	0
16	TEA_SPO...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0
17	TONGS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0
18	WHISK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0
19	WOODEN...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0
20	PIZZA_C...	4	0	0	0	0	0	3	0	1	1	0	0	0	4	0	0	1	0	1	0
21	POTATO ...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0

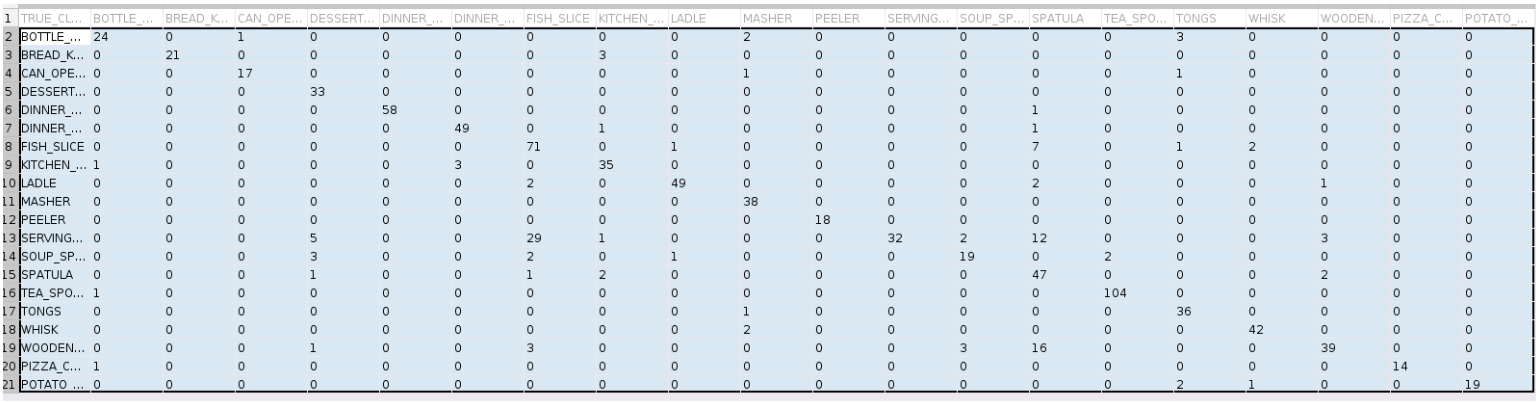
Fig 5.3 Confusion Matrix for Bayes Classifier with ci6 removed.

5.2 Hierarchical Classifier

The classifiers used at each level of hierarchy with feature selection are given below -

- Level 0 – SVM classifier with feature selection to divide the data into left classes and right classes. The accuracy obtained for this division was 98.44% after removing the features ci2 and ci4.
- Level 1 – SVM classifier with feature selection for both the left and right classes at level 1 to further divide the data. The accuracy for division of left classes was 98.26% with all the features and for the right classes it was 97.15% with all the features.
- Level 2 – The four groups of classes at level 2 are classified using a Bayes classifier for each group trained separately with the features created earlier. Feature selection is used to ensure maximum accuracy of these classes at level 2.

The macro accuracy obtained using the above stream-flow was **87.50%** and the micro accuracy is **85.38%.** The confusion matrix for this hierarchical classifier is shown in figure 5.4.



1	TRUE_CL...	BOTTLE_...	BREAD_K...	CAN_OPE...	DESSERT...	DINNER_...	DINNER_...	FISH_SLICE	KITCHEN_...	LADLE	MASHER	PEELER	SERVING...	SOUP_SP...	SPATULA	TEA_SPO...	TONGS	WHISK	WOODEN...	PIZZA_C...	POTATO_...
2	BOTTLE_...	24	0	1	0	0	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0
3	BREAD_K...	0	21	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
4	CAN_OPE...	0	0	17	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
5	DESSERT...	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	DINNER_...	0	0	0	0	58	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
7	DINNER_...	0	0	0	0	0	49	0	1	0	0	0	0	0	1	0	0	0	0	0	0
8	FISH_SLICE	0	0	0	0	0	0	71	0	1	0	0	0	0	7	0	1	2	0	0	0
9	KITCHEN_...	1	0	0	0	0	3	0	35	0	0	0	0	0	0	0	0	0	0	0	0
10	LADLE	0	0	0	0	0	0	2	0	49	0	0	0	0	2	0	0	0	1	0	0
11	MASHER	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0
12	PEELER	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0
13	SERVING...	0	0	0	5	0	0	29	1	0	0	0	32	2	12	0	0	0	3	0	0
14	SOUP_SP...	0	0	0	3	0	0	2	0	1	0	0	0	19	0	2	0	0	0	0	0
15	SPATULA	0	0	0	1	0	0	1	2	0	0	0	0	0	47	0	0	0	2	0	0
16	TEA_SPO...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	0	0
17	TONGS	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	36	0	0	0	0
18	WHISK	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	42	0	0	0
19	WOODEN...	0	0	0	1	0	0	3	0	0	0	0	0	3	16	0	0	0	39	0	0
20	PIZZA_C...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0
21	POTATO ...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	19

Fig 5.4 Confusion Matrix for Hierarchical Classifier

5.3 Discussion

As we did not have time to complete cross-validation experiments, this performance on the training data gives an optimistic estimate of the performance on the new images.

APPENDIX

MATLAB Codes

Following is the list of the MATLAB codes with a short description of each -

1. For computing features

- threshold.m – Computing the binary image for each raw image in each class.
- complexmoment.m – it computes the complex moment invariants for the binary image.
- Prongs.m – Computes the feature prongs described above.
- Convex.m – Computes the feature convex described above.
- skeletonization.m – Computes the skeleton feature described above.
- com.m – Computes the elongation feature described above.
- erosion.m – Computes the erode feature described above.
- holes.m – Computes the Hole feature described above.
- shape.m – Computes the feature shape described above.
- edge.m – Computes the feature edge described above.
- mask.m – Computes the feature colour angle described above.
- symmetry.m – Computes the feature Symmetry described above.
- getproperties.m – Returns a vector after computing all the features for an image.
- create_feature_vector.m – Creates a table with all the features together of an image.
- collect_feature_vectors.m – Collects all the feature vectors for all the images of a given class into a table.
- collect_all_features.m – Creates a table for all the features and for all the classes.
- znorm.m- Computes the normalized feature values from the original feature table.

- histogram_plot.m – Histogram plots for all features with all the 20 classes together in a single feature for comparison.
- plotgauss.m – Gaussian plots for all the features.
- plot_all_gauss.m – Gaussian plots for all the classes.

2. Bayes Classifier [1]

- buildmodel.m – Builds the model for the Bayes classifer using the z-normalized features.
- photocode.m – Provides a code to each class for reference.
- multivariate.m – Multivariate Gaussian classifier used for classification.
- classifier.m – Uses the model build by 'buildmodel.m' to classify any new image.
- create_confusion_matrix.m – Creates a confusion matrix for comparing the performance of the classifier.
- confusion_accuracy.m – Computes micro and macro accuracy from the confusion matrix obtained above.

3. Feature Selection

- feature_selection.m – Returns the subset of features that yields the maximum accuracy for the Bayes classifier.
- feature_svm – Returns the subset of features that yields the maximum accuracy for multiclass SVM/ two-class SVM.
- buildmodel_hier.m – buildmodel.m edited to take into account feature selection while classification.
- classifier_hier.m – classifier.m edited for feature selection.
- create_confusion_matrix_hier – create_confusion_matrix.m edited for feature selection.

4. Hierarchical Classifier

- select_classes.m – Selecting the group of classes that will go to the left and right side of the tree.
- svm_classify.m – Division of classes at level 0 and level 1 of hierarchy, this function is used for classification.
- multiclass_svm.m – For classification of the groups of classes at level 2 of hierarchy.
- main_hier.m – Checking the confusion accuracy of the A:B(A – left classes, B- right classes), A1:A2 and B1:B2.
- test_hier.m – Builds separate feature tables for left and right classes using either select_classes.m as the criteria or hand crafting classes into respective left and right sides.
- accuracy_hier.m – Tests the accuracy of the hierarchical classifier using different classifiers at different steps.

References

[1] D.Fullerton, "*A visual Database of Recognisable Kitchen Utensils",* Undergraduate Dissertation, School of Informatics, University of Edinburgh, 2016.

[2] J.Flusser, B. Zitova, and T. Suk, *Moments and moment invariants in pattern recognition*, John Wiley & Sons, 2009