K-Means & Randomized Forest Codebooks

Munir Suleman and Nat Wannawas
Department of Bioengineering, Imperial College London, London, UK
email: munir.suleman18@imperial.ac.uk, nat.wannawas18@imperial.ac.uk

Abstract – **This work compares K-Means (KMCD) and Randomized Forest (RFCD) Methods on visual codebooks construction. The codebooks were tested on a randomized decision forest (RF) classifier using different parameters.**

The results showed that the training and testing times were slower for the KMCD. Its vector quantization time was faster than the RFCD with a tree depth below 7, but slower with tree depth above 10. The optimal RF classifier parameters were 500 trees with a depth of 7 giving an accuracy of 78.13% for the KMCD, and 700 trees at a depth of 8 giving an accuracy of 68.53% using the RFCB.

I. Introduction

K-means (KMCD) and random forest (RFCB) are common methods used to build visual codebooks. This work compares both methods by building the codebooks and then testing them on a randomized decision forest (RF) classifier with different parameters to optimize accuracy and time efficiency.

Caltech 101 data set with 10 classes is used. 15 training and 15 testing images are randomly selected from each class. SIFT is used as a feature descriptor.

II. K-Means Codebook (KMCB)

A. Vector Quantisation

100,000 patches in the training set are randomly selected from all patches to form a specified amount of cluster centers and are then computed by the '*kmeans*' MATLAB function. All patches of images in training and testing sets are matched to their nearest cluster centers by the '*knnsearch*' MATLAB function. The '*histcounts*' MATLAB function is used for counting the nearest cluster centers to build the bag of words.

B. Vocabulary Size

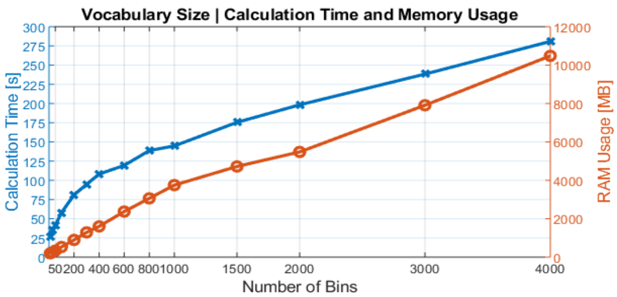

Figure 1: RAM and calculation time used in vector quantization process using the k-means method

Effect of number of bins on memory and time usage for vector quantization is shown in Figure 1. RAM is recorded during the operation of '*kmeans'* function which is the highest usage during the process. Both time and memory grow nearly linearly with number of bins.

C. Bag of Words Histograms

Figure 2 shows example of histograms (bag of words) obtained from the KMCB. Histograms of an image are different with different number of bins (vocabulary size). However, with the same number of bins, images from the same class share some similar features. Each class has their own distinctive features. For example, with 10 bins, both train and test images of the wild-cat class have twin bars around 600 patches in height on codewords no. 7 and 9.

Similarity shared among their own classes and differences across different classes influence accuracy of classification. If the applied vocabulary size provides high similarity within classes and differences across classes, it will have a higher likelihood to be classified correctly.

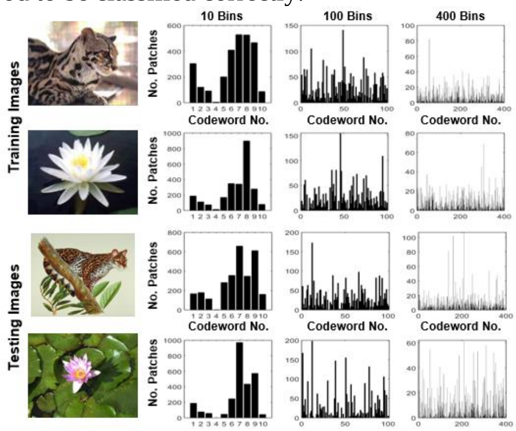

Figure 2: Example of histograms of training and testing images obtained through the k-means method

III. Random Forest Codebook (RFCB)

A. Vector Quantisation

667 patches per image (100,050 patches total), were randomly selected in the training set. The patches were used to grow the trees using several parameters (Table 1) and the axis-aligned weak learner. All patches of images were passed through the trees, using '*histcounts*' function to get the number of patches arriving at each leaf to build the bag of words.

B. Vocabulary Size

Vocabulary size obtained from different number of trees and tree depth with 10 split trials are shown in Table 1. Vocabulary size of the RFCB grows linearly with number of trees and exponentially with tree depth. The results from 20 split trials are insignificantly different (see Appendix I).

In terms of time, Figure 3 shows the vector quantization time is linearly proportional to vocabulary size, with their rate (shown by the slope) inversely depending on tree depth.

Table 1: Vocabulary size obtained from RF with 10 split trials

The number of codewords (Bins) *with 10 numSplits*

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	4	20	40	80	120	200	400
5	16	80	160	318	479	799	1596
7	64	309	610	1268	1876	3155	6311
10	437	1996	4116	8538	13134	21374	42995
12	1383	6189	11790	24569	36954	59760	123274

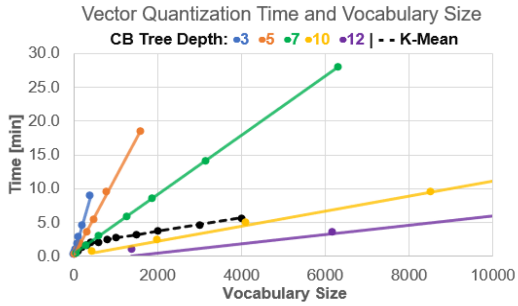

Figure 3: Time used in vector quantization against vocabulary size

C. Bag of Words Histograms

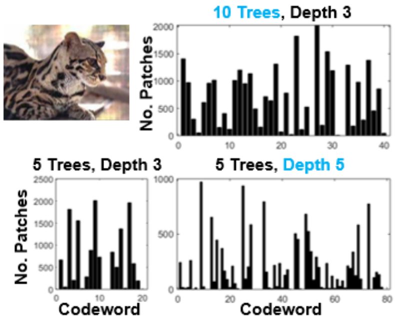

Figure 4: Example histograms of an image from RF codebook.

Figure 4 shows examples of histograms of an image obtained with the random forest codebook with its parameters.

IV. Randomized Decision Forest Classifier

The training images were used to grow the trees by a specified split function. The testing images were then passed through the forest and categorized into the leaves which correspond to their predicted class.

There are three main parameters involved within the classifier that required optimizing: Number of trees, number of split trials and the depth of the tree. The number of bins for the codebook was also considered in the optimization process. The iteration method (i.e. only changing one parameter at a time and using the best value for that parameter for the next step) is shown in Figure 5. Axis-aligned and two-pixel test weak learners were both tested to see which provided the better results in terms of accuracy and run time. This section details the results obtained by both codebooks and compares them.

A. K-Means Codebook

Firstly, we optimized the tree depth (Figure 5a), and number of split trials (Figure 5b) with 100 trees. These gave the initial optimal number of 10 for both depth and split trials. These numbers are used as the base parameters. The other parameters are then fine-tuned to find the optimum values. The optimum parameters for both the axis-aligned and two-pixel test weak learners are shown in Table 2.

Table 2: Optimum parameters and the corresponding accuracy and time for both codebooks and weak-learners

	No. of Bins	No. of Trees	No. Split Trials	Tree Depth	Accuracy (%)	Time (sec)
K-Means: AA	400	500	10	7	78.13	67
K-Means: PT	400	500	10	10	76.13	73
Random Forrest: AA	610	700	10	9	67.86	69
Random Forrest: PT	610	700	10	8	68.53	50

AA = Axis-Aligned, PT = Two-Pixel Test

Figure 5 shows how the accuracy and time varied in relation to the parameter being changed. For both cases the relationship is the same. For all parameters, the accuracy increased up to a point where it eventually plateaus. The variance for the accuracy when changing all parameters is high, showing possibly how unreliable the classifier can be.

Due to the plateau, time then becomes the deciding factor for the optimum value for the parameters. Besides the tree depth, the increase in time is linear. Figure 5 a and d shows an exponential increase in time as tree depth increases, which signifies how important choosing a low tree depth with a high accuracy is. Table 2 shows that for both weak learners the optimum parameters are mostly the same, apart from the tree depth, which is higher for the two-pixel test.

The axis aligned with the KMCB is shown to have the higher accuracy by 2% and has the lower run time only due to the lower tree depth. Thus, it can be deduced the two-pixel test is a faster weak learner than the axis aligned. Besides this, the axis-aligned method is a better weak learner when using the KMCB.

B. Random Forest Codebook

Data obtained from the RFCB are tested on a RF classifier with base parameters, axis-aligned and two-pixel test weak learners. The results of axis-aligned case are shown in Table 3. For both weak learners, the highest accuracy was at 10 trees and at a depth of 7, similar to the RFCB (Appendix II). These parameters were chosen for further optimization. Result is shown in Table 3 (and in Appendix III).

Table 3: Accuracy of the data from RF codebook

Acc. [%]	Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
	3	43	43	40	52	50	50	51
	5	41	59	57	54	59	57	54
	7	51	57	61	56	52	55	53
	10	48	47	50	53	50	48	50
	12	47	47	48	46	49	47	50

C. Classification Successes and Failures

Table 4 shows the confusion matrix for the k-means, two-pixel test method with optimal parameters. Both codebooks and weak learners most commonly misclassified are the umbrella and wild-cat (see Appendix IIII). In both cases they are misclassified as a wrench often.

Figure 6 shows examples of histograms, where images were correctly or incorrectly classified. In the cases of the correctly

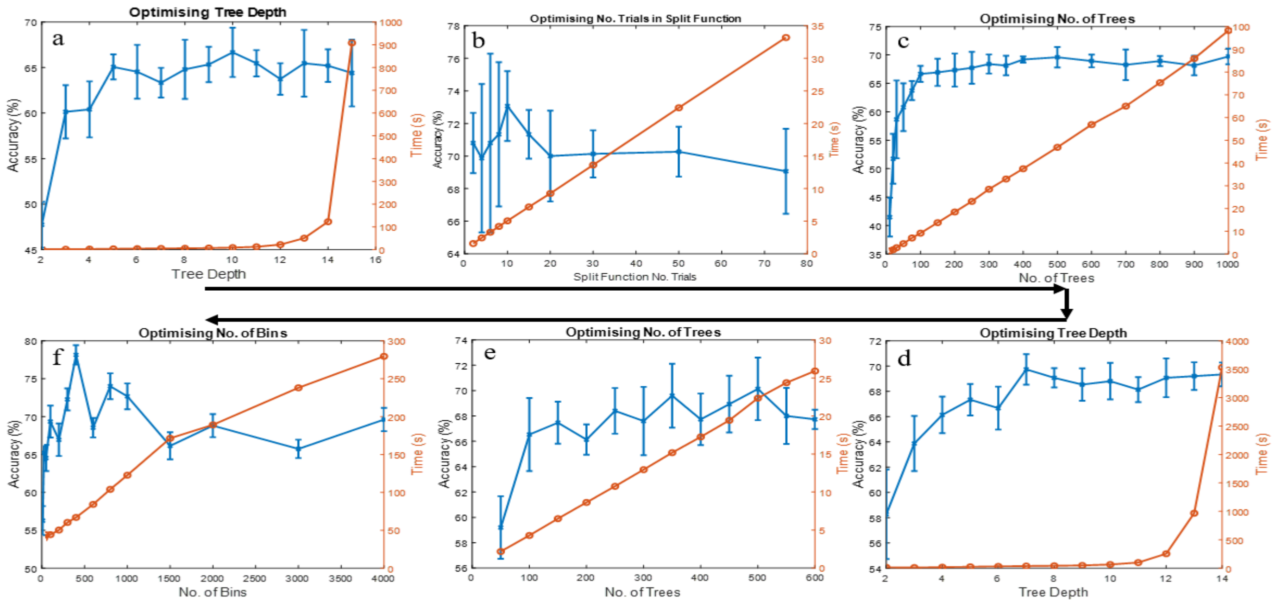

Figure 5: The optimization process for the parameters. The graphs show how the accuracy and time are affected by varying a single parameter

classified wild-cat and water-lily, the peaks in the histograms (at approx. 50 and 300 codewords respectively) match that of the training images shown earlier in Figure 2. However, for the misclassified images, the peaks in the histograms resemble that of the training images of a wrench (at approximately 200 codewords), which explains why it was classified as such.

Table 4: Confusion matrix of the KMCB, axis-aligned method

	Classified Label									
Class Names (True Label)	1	2	3	4	5	6	7	8	9	10
1	59	0	0	0	5	5	3	2	1	0
2	0	75	0	0	0	0	0	0	0	0
3	1	0	34	0	11	5	4	4	16	0
4	3	0	0	42	13	7	2	0	8	0
5	0	2	0	2	60	5	6	0	0	0
6	2	0	0	1	1	56	11	4	0	0
7	5	4	0	3	7	0	44	0	12	0
8	0	0	0	0	0	4	0	71	0	0
9	0	0	5	0	3	0	1	0	65	1
10	9	0	0	0	0	0	0	1	0	65

Class Names: 1. Tick, 2. Trilobite, 3. Umbrella, 4. Watch, 5. Water Lilly, 6. Wheel-Chair, 7. Wild Cat, 8. Windsor Chair, 9. Wrench, 10. Yin-Yang

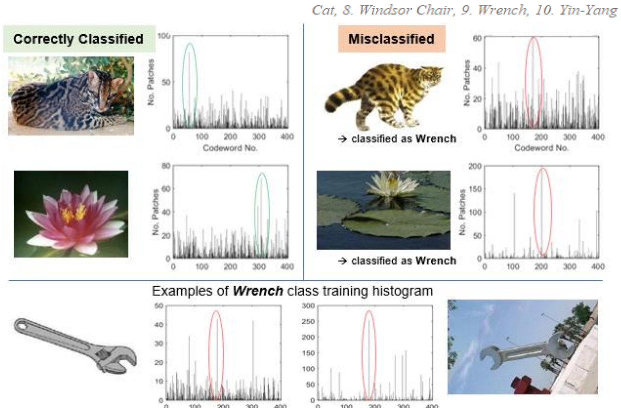

Figure 6: Top-left: successful classification. Top-right: misclassification. Bottom: training images histograms in wrench class

D. Comparing Codebooks

Time used in training and testing of the data from both codebooks grows exponentially with vocabulary size (Figure 7). The KMCB takes longer but gave a higher accuracy across range of vocabulary size. The optimal vocabulary sizes are around 400 to 610 for KMCB and RFCB respectively.

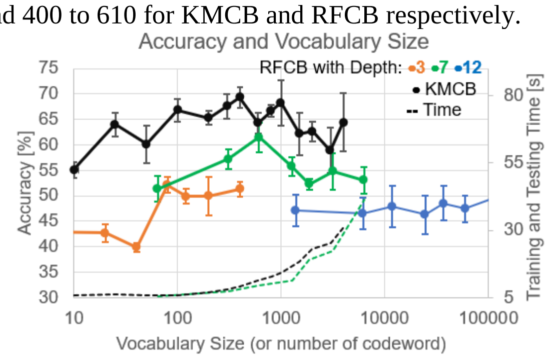

Figure 7: accuracy and time used in training and testing of the data from RFCB and KMCB tested on RF classifier with base parameters

Using the optimal vocabulary size, different RF parameters were required. The RFCB using two-pixel test required more trees a slightly higher tree depth for the best accuracy compared to the KMCB. In terms of training and classifying time, the RFCB takes 50s with a larger vocabulary and classifier size, while the KMCB takes 67s. This is similar to [1], where time taken by RFCB is significantly lower. However, the KMCD gives a higher accuracy which goes against [1].

V. Conclusion

From the data presented above, it can be concluded that KMCB, when comparing at the same vocabulary size, is quicker than shallow RF but slower than RFCB with depth above 10. However, RFCB is quicker in training and classifying.

For the accuracy, the KMCB with the axis aligned weak learner presented the best accuracy of 78.13%. While, two-pixel weak learner gives the best accuracy of 68.53% for RFCB data.

References

[1] F Moosmann et.al, "Fast Discriminative Visual Codebooks," Advances in Neural Information Processing Systems 19, 2006

VI. Appendix I

Vector Quantization Time [minute] — 10 Split Trials

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	0.3	0.6	1.0	1.9	2.9	4.5	8.9
5	0.3	1.0	1.9	3.6	5.4	9.4	18.5
7	0.5	1.6	3.0	5.8	8.6	14.0	28.0
10	0.7	2.5	4.9	9.5	13.9	23.7	48.0
12	1.0	3.6	7.2	14.7	25.1	39.4	84.1

Number of codewords (Bins) — 20 Split Trials

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	4	20	40	80	120	200	400
5	16	80	160	320	480	800	1600
7	64	313	639	1272	1909	3175	6367
10	480	2277	4510	8897	13518	22511	44987
12	1617	6231	13918	26280	39770	68016	138679

Vector Quantization Time [minute] — 20 Split Trials

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	0.4	1.1	2.1	3.9	5.9	9.9	20.2
5	0.6	2.1	4.0	7.7	11.7	19.4	37.9
7	0.7	3.1	5.9	11.4	17.0	28.8	56.7
10	1.1	4.6	9.3	18.7	28.0	45.3	90.9
12	1.4	6.5	13.2	27.1	41.0	68.6	139.7

Figure 8: Number of codewords (vocabulary size) of randomized forest codebook with 1 to 100 trees, 3 to 12 tree depths, 10 and 20 split trials, and axis-aligned weak learner

VII. Appendix II

Accuracy (Axis-Aligned) [%]

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	43	43	40	52	50	50	51
5	41	59	57	54	59	57	54
7	51	57	61	56	52	55	53
10	48	47	50	53	50	48	50
12	47	47	48	46	49	47	50

Accuracy (Two Pixel-Test) [%]

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	38	45	37	51	48	48	47
5	43	59	56	55	59	59	53
7	51	59	61	57	51	56	56
10	50	48	54	56	51	53	54
12	52	48	51	49	53	53	51

Training and Testing Time [minute]

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	0.1	0.1	0.1	0.1	0.1	0.1	0.2
7	0.1	0.1	0.2	0.2	0.3	0.4	0.7
10	0.1	0.3	0.6	1.3	2.2	3.5	7.2
12	0.2	0.8	1.6	3.6	5.6	9.1	18.8

Training and Testing Time [minute]

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	0.1	0.1	0.1	0.1	0.1	0.2	0.2
7	0.1	0.1	0.2	0.2	0.4	0.4	0.8
10	0.2	0.4	0.7	1.5	2.2	3.5	7.3
12	0.2	0.8	1.6	3.7	5.8	10.8	22.7

STD of Accuracy

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	1.1	1.7	0.9	1.6	1.4	3.8	1.4
5	2.9	1.8	1.5	2.2	2.7	0.7	2.6
7	2.5	1.8	3.0	1.9	1.0	3.6	2.6
10	4.2	3.1	1.9	1.4	1.3	5.1	2.9
12	3.1	3.1	4.0	3.7	3.4	2.6	2.7

STD of Accuracy

Tree Depth	The Number of Codebook Tree: 1	5	10	20	30	50	100
3	2.2	2.4	2.4	1.5	1.3	2.6	1.4
5	2.2	0.8	1.2	2.0	1.4	1.2	1.5
7	3.2	3.1	3.1	2.1	2.6	1.0	2.4
10	1.5	3.7	2.2	1.2	3.0	4.0	1.9
12	1.5	2.6	1.4	1.5	4.8	2.8	5.2

Figure 9: accuracy and training and testing time of the data from RFCB with 10 trials. 100-tree RF with tree depth of 10 used. The numbers shown are the average across 5 trials

VIII. Appendix III

Accuracy (Axis-Aligned) [%]

Tree Depth	Number of Trees: 100	300	500	700
5	56.13	60.40	62.27	62.67
6	56.93	61.07	63.20	64.13
7	56.27	62.80	64.13	65.33
8	57.33	64.67	66.40	66.80
9	59.33	62.93	65.73	67.87
10	59.20	64.53	66.53	66.80
11	60.67	65.33	67.47	67.47
12	59.20	64.80	67.73	67.73

Accuracy (Two-Pixel Test) [%]

Tree Depth	Number of Trees: 100	300	500	700
5	57.33	62.27	63.20	64.13
6	58.27	62.40	63.73	64.40
7	60.93	63.73	64.67	65.60
8	61.87	64.67	65.33	68.53
9	60.53	64.00	66.00	65.60
10	62.40	66.27	67.20	67.60
11	61.47	64.80	65.33	65.73
12	60.80	65.07	66.13	66.13

STD (Axis Aligned) [%]

Tree Depth	Number of Trees: 100	300	500	700
5	4.48	1.38	1.30	1.15
6	4.91	1.67	2.80	1.19
7	1.67	2.76	2.47	1.25
8	2.31	1.70	1.53	1.73
9	2.87	1.67	1.12	1.59
10	3.69	2.02	1.28	1.10
11	2.87	1.89	1.45	1.52
12	2.88	1.45	1.61	1.38

STD (Two-Pixel Test) [%]

Tree Depth	Number of Trees: 100	300	500	700
5	2.49	1.92	2.08	0.99
6	1.61	2.56	1.98	1.53
7	1.21	2.52	2.45	1.01
8	2.08	1.49	1.70	1.28
9	1.28	1.70	2.83	1.53
10	1.80	2.39	2.08	1.74
11	2.96	1.28	2.40	1.21
12	1.97	0.76	1.79	0.73

Time (Axis-Aligned) [minute]

Tree Depth	Number of Trees: 100	300	500	700
5	0.06	0.21	0.36	0.47
6	0.08	0.26	0.44	0.61
7	0.11	0.32	0.53	0.73
8	0.12	0.37	0.61	0.86
9	0.16	0.48	0.81	1.15
10	0.20	0.65	1.04	1.47
11	0.30	0.96	1.67	1.81
12	0.44	1.46	2.55	4.08

Time (Two-Pixel Test) [minute]

Tree Depth	Number of Trees: 100	300	500	700
5	0.06	0.18	0.30	0.42
6	0.08	0.23	0.39	0.55
7	0.10	0.31	0.51	0.71
8	0.12	0.35	0.59	0.83
9	0.15	0.44	0.78	1.08
10	0.18	0.56	0.95	1.36
11	0.26	0.80	1.35	1.97
12	0.45	1.44	2.61	4.26

Figure 10: Optimization Parameter of RF classifier for the data of RFCB with 100 trees and tree depth of 7

IX. Appendix IIII

K-means - PT

True Label / Class Names	Classified Label: 1	2	3	4	5	6	7	8	9	10
1	59	0	0	0	5	5	3	2	1	0
2	0	75	0	0	0	0	0	0	0	0
3	1	0	34	0	11	5	4	4	16	0
4	3	0	0	42	13	7	2	0	8	0
5	0	2	0	2	60	5	6	0	0	0
6	2	0	0	1	1	56	11	4	0	0
7	5	4	0	3	7	0	44	0	12	0
8	0	0	0	0	0	4	0	71	0	0
9	0	0	5	0	3	0	1	0	65	1
10	9	0	0	0	0	0	0	1	0	65

RFCD - Axis Aligned

True Label / Class Names	Classified Label: 1	2	3	4	5	6	7	8	9	10
1	40	7	0	2	0	21	5	0	0	0
2	0	71	0	1	0	0	0	3	0	0
3	0	0	3	6	10	6	5	0	25	20
4	0	0	0	54	4	7	0	3	7	0
5	3	3	0	1	46	7	3	0	10	2
6	0	0	0	6	0	64	5	0	0	0
7	6	4	0	11	14	6	34	0	0	0
8	3	0	0	0	0	3	0	69	0	0
9	0	0	0	0	0	0	0	0	71	4
10	10	2	3	0	3	3	2	1	0	51

RFCB - Two-Pixel Test

True Label / Class Names	Classified Label: 1	2	3	4	5	6	7	8	9	10
1	36	8	0	0	0	21	10	0	0	0
2	0	71	0	0	0	0	0	4	0	0
3	0	2	0	6	14	5	3	0	25	20
4	0	0	0	55	3	8	0	3	6	0
5	0	5	0	2	51	7	0	0	5	5
6	0	0	0	3	2	65	5	0	0	0
7	5	4	0	11	15	5	33	0	2	0
8	0	0	0	0	0	2	0	73	0	0
9	0	0	0	0	1	0	1	0	67	6
10	4	14	0	2	2	3	1	2	0	47

Class Names: 1. Tick, 2. Trilobite, 3. Umbrella, 4. Watch, 5. Water Lilly, 6. Wheel-Chair, 7. Wild Cat, 8. Windsor Chair, 9. Wrench, 10. Yin-Yang

Figure 11: Top: Confusion matrix for the KMCD Two-pixel test. Middle: Confusion Matrix RFCD Axis-aligned. Bottom: Confusion Matrix RFCB two-pixel test