Shape Matching and Retrieval

Shape based matching for object recognition is a more robust method than color/texture based matching. Majority of the information required for image recognition is already present in the shape contours. There exists several shape based methods for matching. The basic guidelines behind a robust shape based matching method are-

- Extract shape features
- Match across images in a deformable way
- Representation and matching should be compact and efficient
- Should be able to deal with unseen textures, poses, deformation etc.

In this project, we have shown a comparative study of the following methods for Shape Matching and Retrieval-

1. **Shape Context descriptors based matching** [1]
2. **Chamfer distance based matching**[2]

Dataset Used - MPEG-7 dataset: MPEG-7 is a popular dataset used for shape matching. It consists of 70 shape classes with 20 different images in each class with high intra-class variations. Each image are a binary mask of an object of the respective class.

We select one image from each of the class as a template image and match each template with the rest of the images in the database. We have shown Precision@20 for each of the methods.

1 Shape Context Descriptors based matching

Idea of this technique is to convert a particular shape into a vector/descriptor and then perform matching of these descriptors. For each point on the edge map of the shape we compute a descriptor which counts the number of pixels present at different angles and distances. There are 5 radial bins and 12 angular bins i.e. the descriptors are of 60 dimensions.

Procedure Followed:

1. **Step 1:** Randomly select some points on the shape contours of the images to be matched.
2. **Step 2:** Compute shape context descriptors for each of these points.
3. **Step 3:** Compute cost matrix between the descriptors of a pair of images. Minimize the matching cost using algorithms for min cost bipartite matching problem like Hungarian Algorithm[3].
4. **Step 4:** Run the algorithm for all the template and image pairs and find the best match for each of the template.

1.1 Matching Results:

Below we have shown how points of template image are matched with that of top ten retrieved shapes from the dataset.

Table 1: Top Ten retrieval results on 'apple-1' template shape

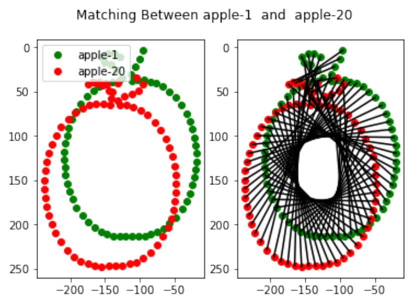


Figure 1: Cosine diff: 0.95 and Std diff: 17.51

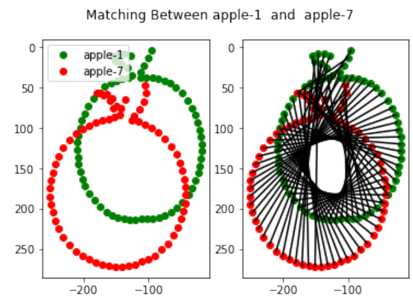


Figure 2: Cosine diff: 0.95 and Std diff: 16.99

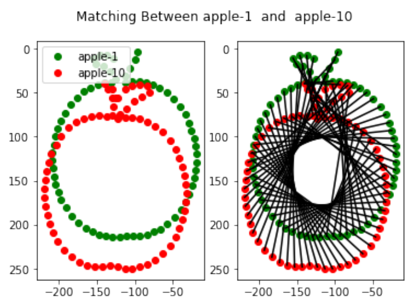


Figure 3: Cosine diff: 0.89 and Std diff:18.55

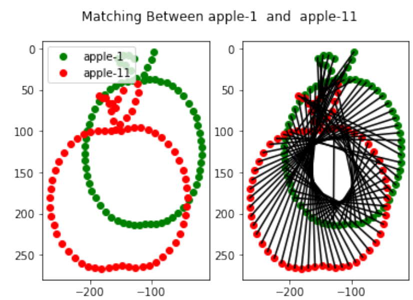


Figure 4: Cosine diff: 0.95 and Std diff: 19.37

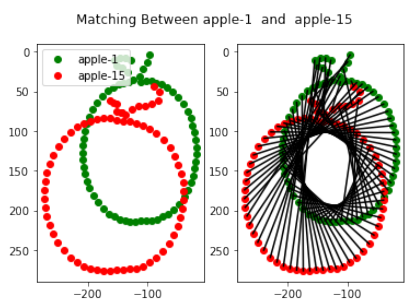


Figure 5: Cosine diff: 0.92 and Std diff: 17.23

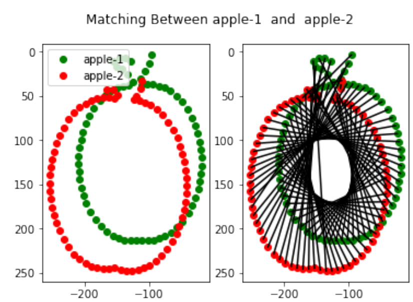


Figure 6: Cosine diff: 0.93 and Std diff: 11.27

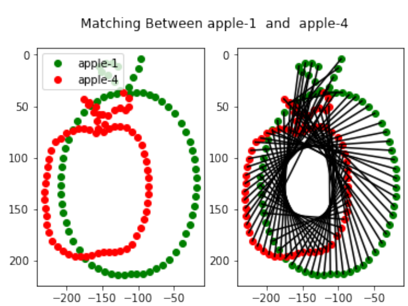


Figure 7: Cosine diff: 0.91 and Std diff:18.23

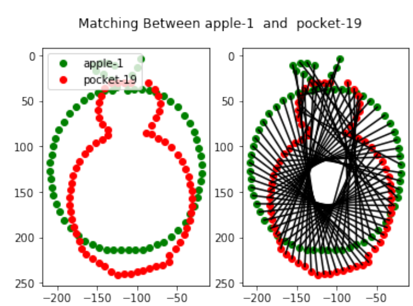


Figure 8: Cosine diff: 0.95 and Std diff: 17.13

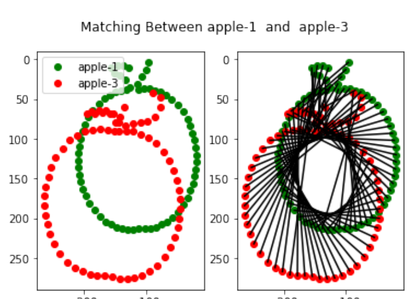


Figure 9: Cosine diff: 0.94 and Std diff: 19.97

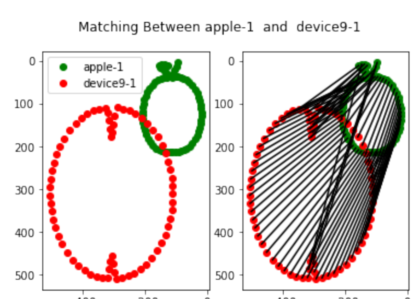


Figure 10: Cosine diff: 0.19 and Std diff: 8.96

Table 2: Top Ten retrieval results on 'beetle-1' template shape

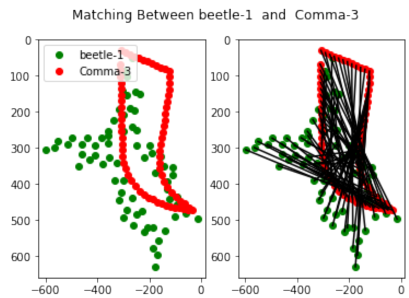


Figure 1: Cosine diff: 0.87 and Std diff: 16.92

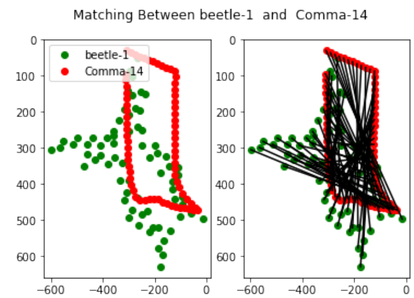


Figure 2: Cosine diff: 0.87 and Std diff: 17.20

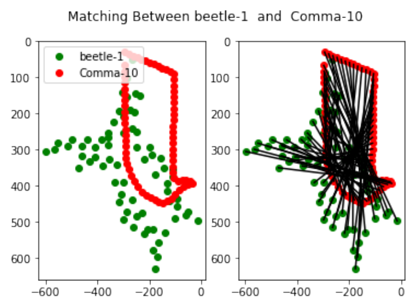


Figure 3: Cosine diff: 0.89 and Std diff:18.85

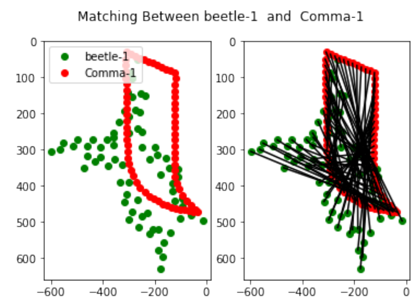


Figure 4: Cosine diff: 0.88 and Std diff: 18.33

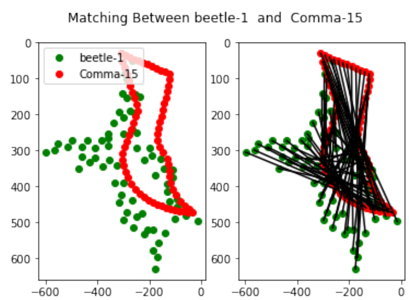


Figure 5: Cosine diff: 0.88 and Std diff: 16.53

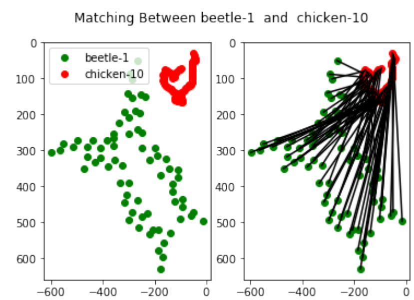


Figure 6: Cosine diff: 0.86 and Std diff:19.75

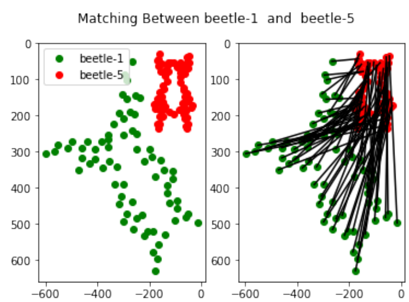


Figure 7: Cosine diff: 0.56 and Std diff:26.16

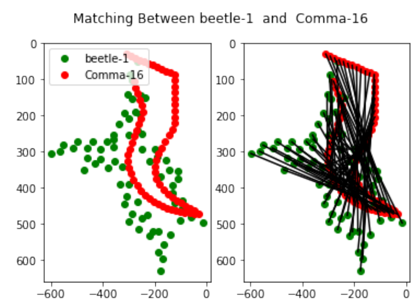


Figure 8: Cosine diff: 0.85 and Std diff: 16.02

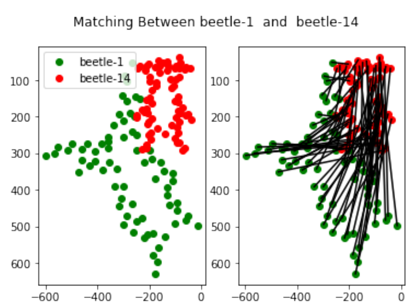


Figure 9: Cosine diff: 0.65 and Std diff: 24.51

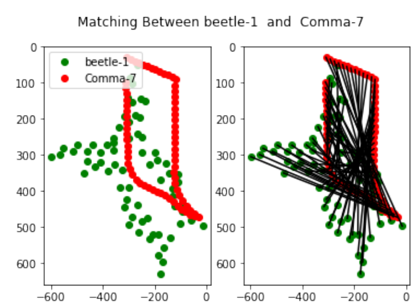


Figure 10: Cosine diff: 0.86 and Std diff: 18.18

1.2 Retrieval Results:

We experimented with two costs functions- **chi-sqaure distance** and **cosine distance**.

Cost Functions	Precision@20
Chi-Square distance	0.55
Cosine distance	0.75

Table 3: Precision@20 on MPEG-7 dataset

Below we have shown retrieved shaped for two template images-

- **Template Image: 'apple1.fig'**

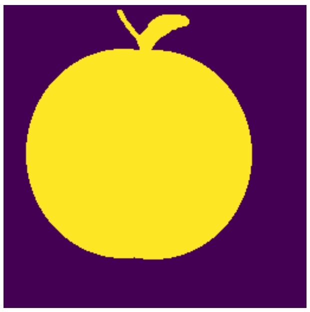


Table 4: Cost Function: Chi-Square distance

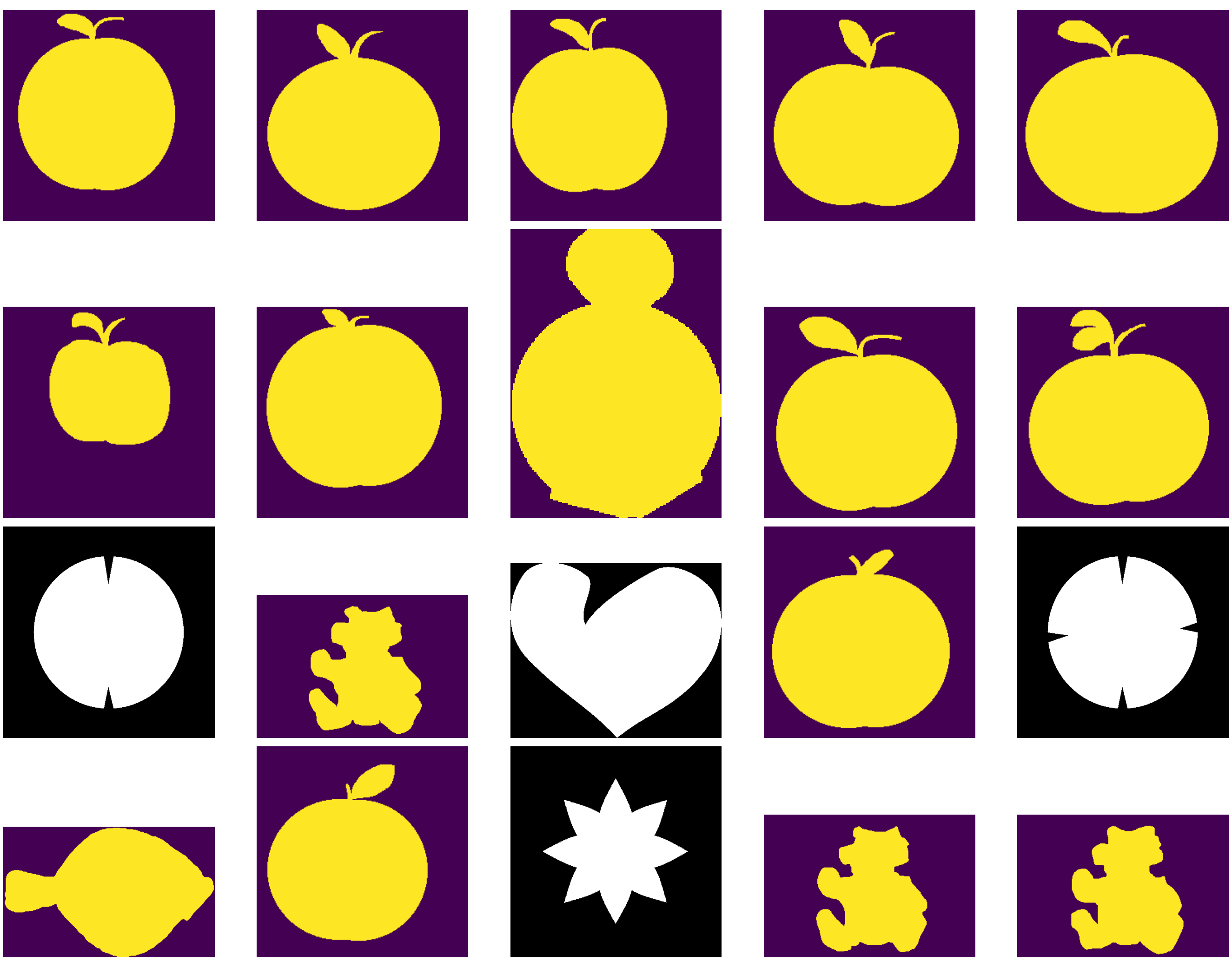
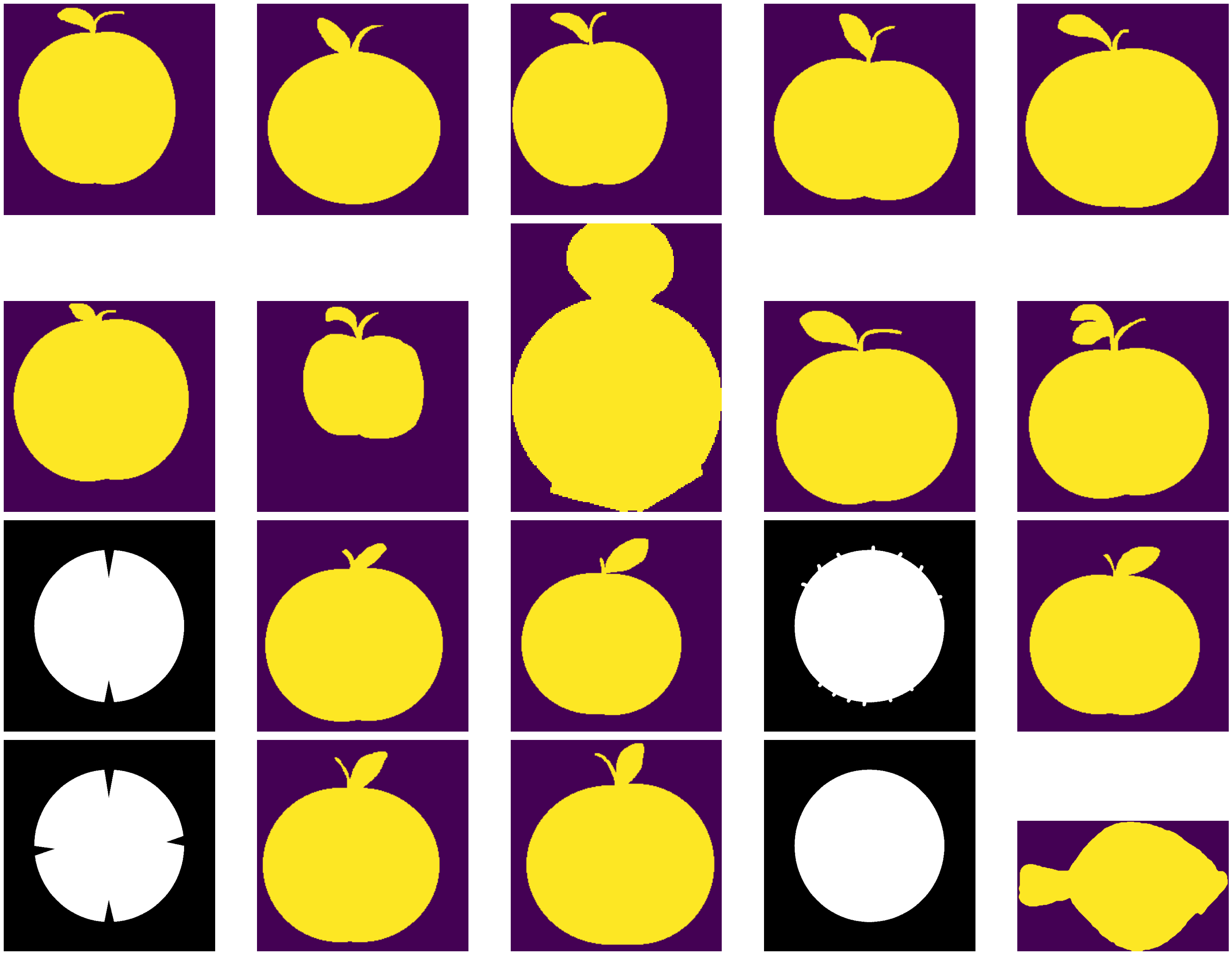


Table 5: Cost Function: Cosine distance



Observation: We observe that for 'apple' shape template retrieved results are impressive. All the retrieved shapes are mostly of an apple except two shapes- pocket and device. But these shapes are of almost similar structures as the template image.

- **Template Image: 'beetle1.fig'**

- **Observation:** In case of 'beetle' shape template the retrieved results mostly contains 'comma' shape. This could be because the shape of 'beetle' is very complex and also may be because the shape of 'comma' is almost similar to some parts(legs) of the template shape(beetle).

Table 6: Cost Function: Cosine distance

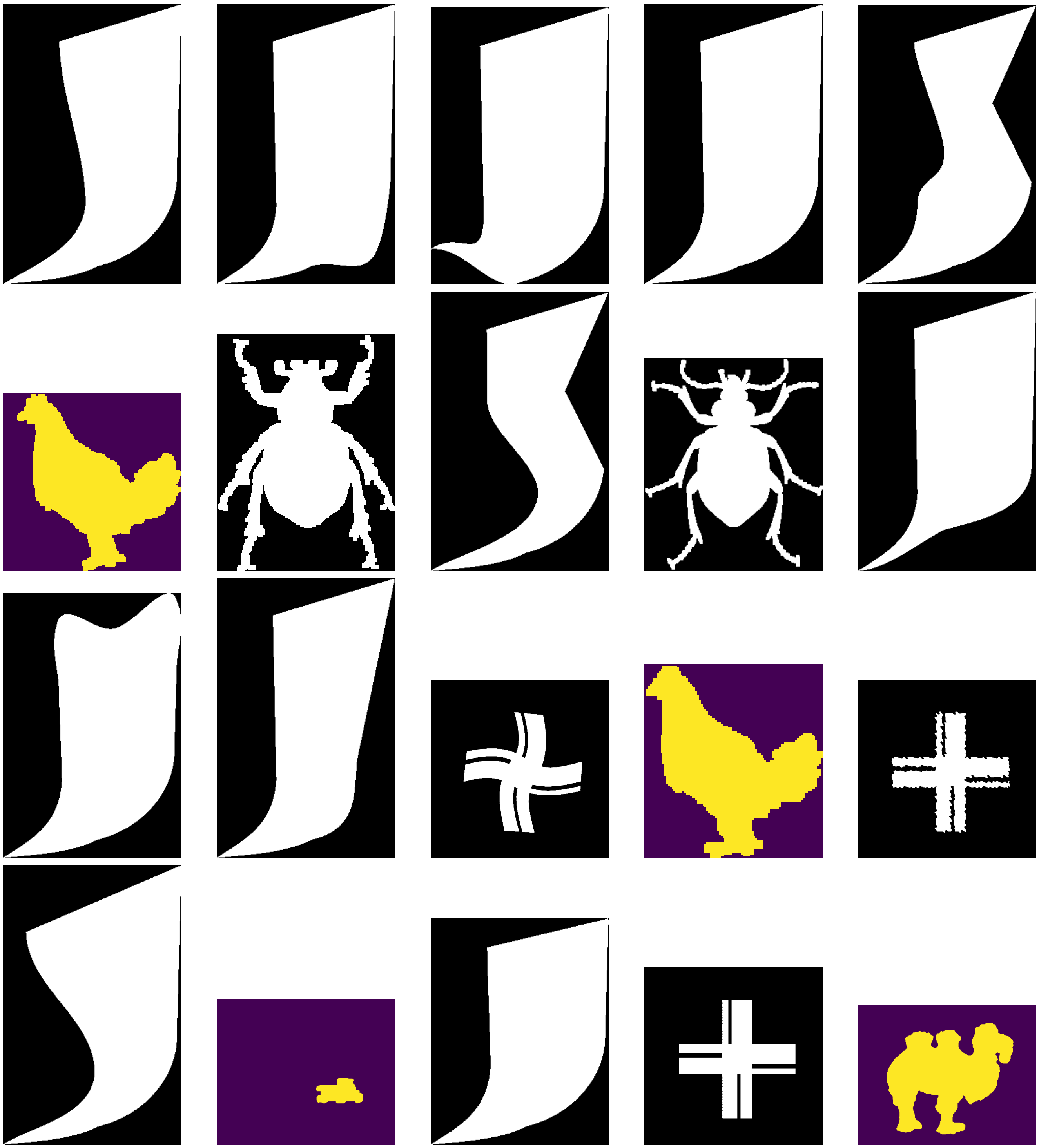
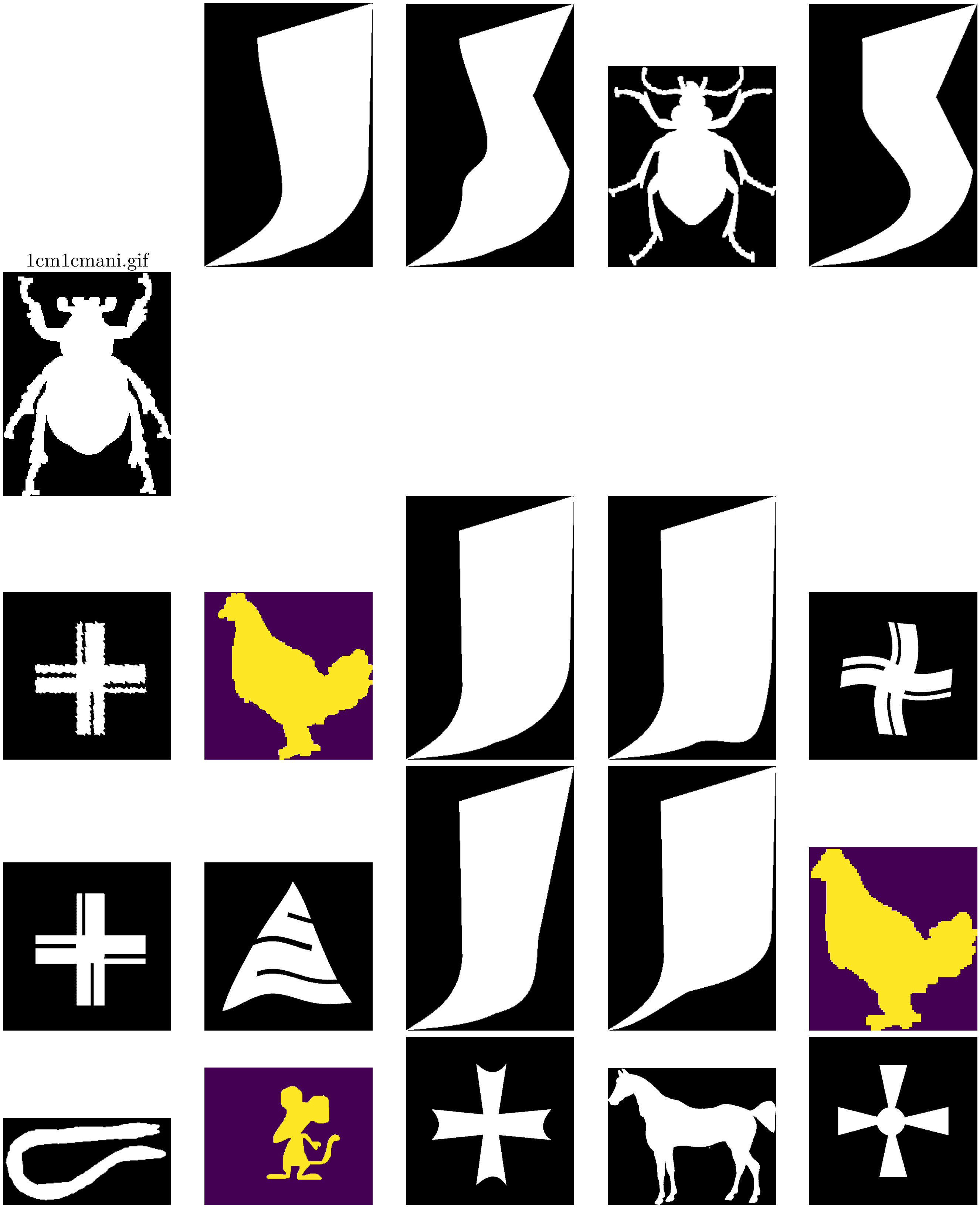


Table 7: Cost Function: Chi-Square distance



1.3 Classification Results:

To check how good are these descriptors in capturing the shape information we use them in shape classification. For this, we combine Shape context descriptors for some randomly selected points on the contour of a shape into one. We take this as a feature vector of a shape. Similarly, we obtain the feature vectors of all the shapes in the dataset. We divide the dataset into train and test set and train the SVM on the feature vectors of the train set. We achieve an accuracy of **81.64%** on the test set.

2 Chamfer Distance based Matching

2.1 Theory

Chamfer Distance is the distance of a pixel from the nearest edge pixel. This distance can be L1 distance. By computing the distance transformation on the query image, the edgemap of template image is overlaid on top of the transformed image.

2.2 Procedure

The procedure is as follows:

1. Choose an image as a template image and rest of the images will be query images.
2. Compute edgemaps of template and query images.
3. Determine the edge points coordinates for the template image.
4. For the query image, using the edgemap compute the Chamfer / L1 distance transformation on the image. This can be computed with Forward and Backward Pass algorithm.
5. To compute distance transformation, initialize the edge points with zero and rest of the points with infinity.
6. In the forward pass, the pixel values is Min(current_value, left_pixel_value + 1, top_pixel_value + 1)
7. In the backward pass, the pixel values is Min(current_value, right_pixel_value + 1, bottom_pixel_value + 1)
8. After computing distance transformation, sum over the pixels which are edge points in the template image.
9. Do this for different scales, rotations, translations and flips of the template image.
10. The minimum of these values gives the Chamfer distance between that template and query image.
11. Repeat this for all query images in the dataset.
12. Repeat the above for all template images or edgemaps.

2.3 Experiment Results

1. The Crown Template Edgemap

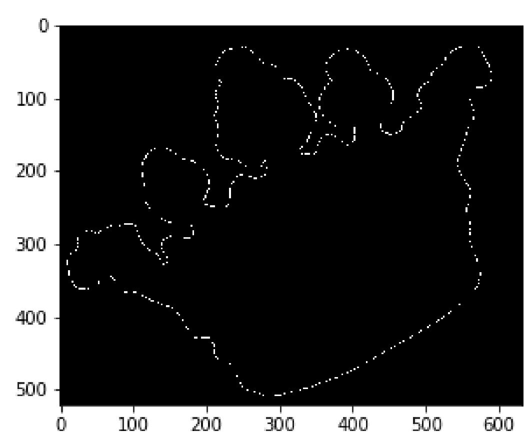
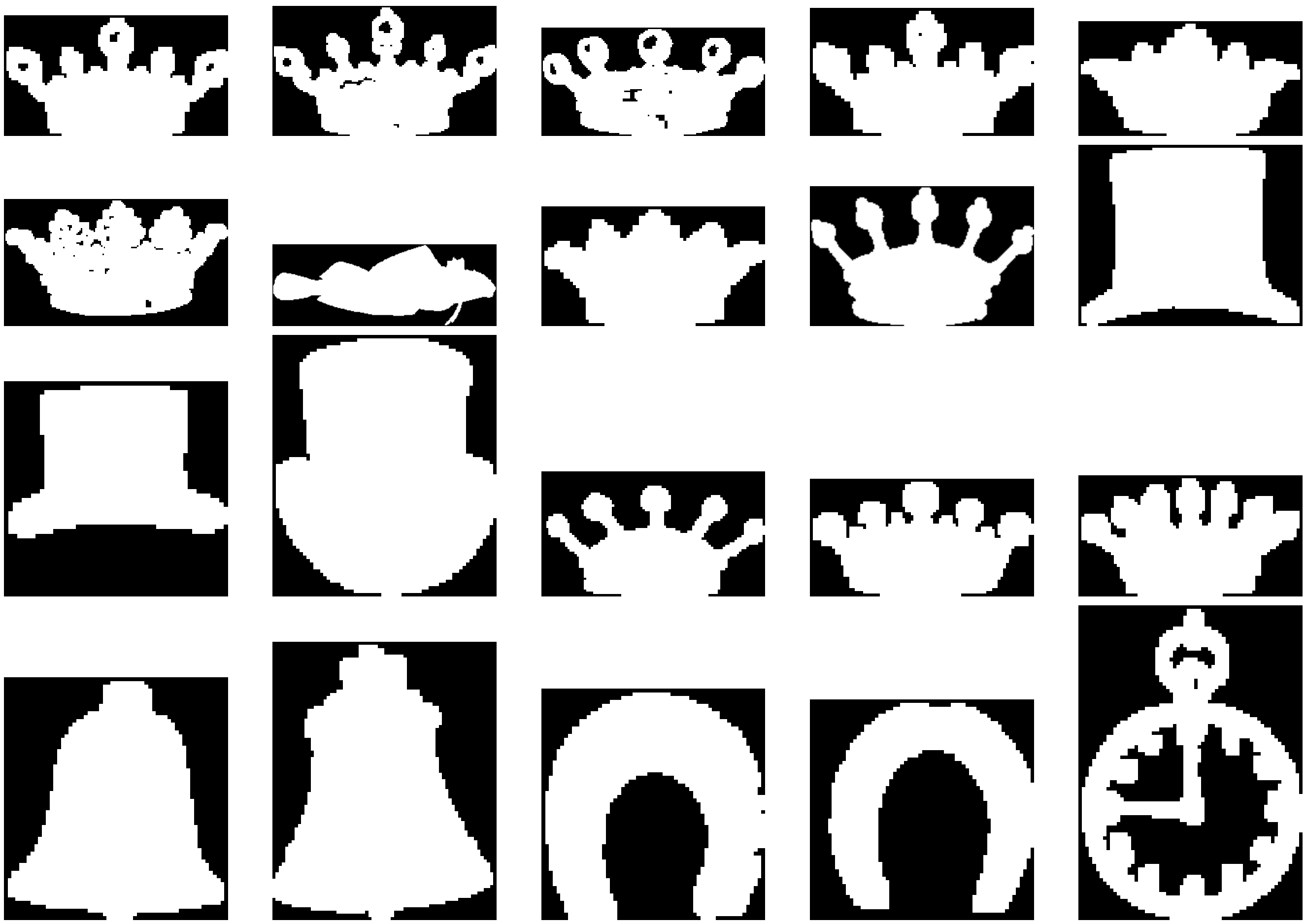


Table 8: Retrieval Result for Crown



2. Beetle Template Edgemap

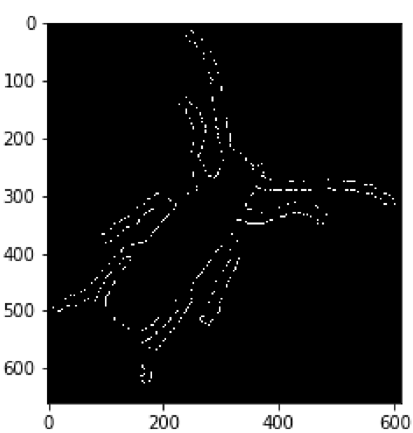
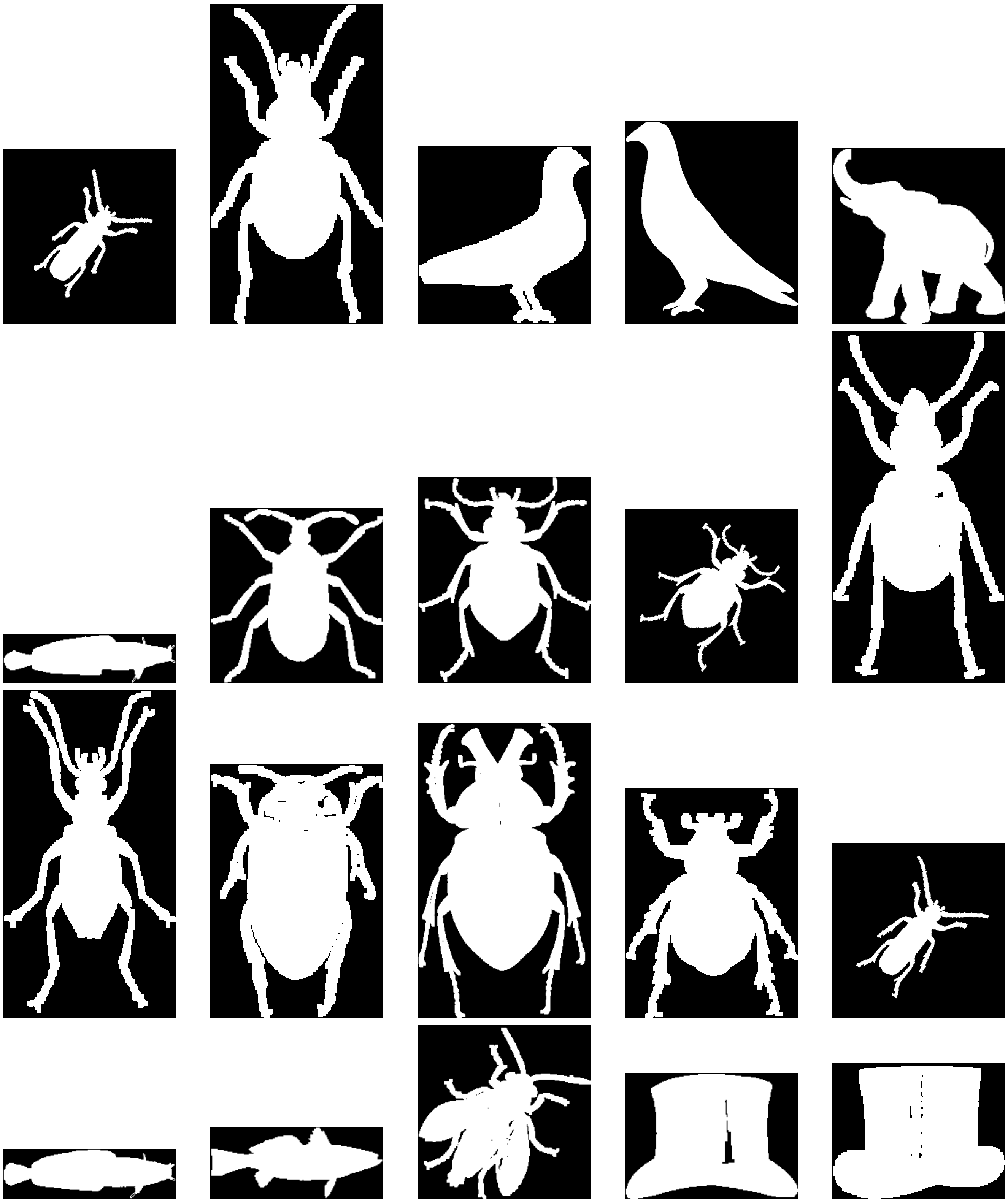


Table 9: Retrieval Results for Beetle



2.4 Observations

- Due to Data Augmentation of template image, quite good retrieval results are obtained.
- Some of the retrievals which are mismatched look similar to the template.

- For example, with Crown template, hat is also retrieved in top 20 matches because of its similarity in the shape. Similarity can be also observed with Bell and pocket watch.
- In the case of Beetle template, many of the beetle queries are correctly matched.
- Some of the retrievals include flies and birds which share similar trends as of beetles.

2.5 Results on Dataset MPEG-7

The Precision@20 score was found out to be around **0.55** for the entire MPEG-7 dataset considering one template per class.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in neural information processing systems*, pages 831–837, 2001.

[2] Gunilla Brogefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):849–865, 1988.

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.