Project general description

- Implement an object recognition pipe.
- Train and tune it on a subset of the Caletch101 dataset.
- Test the accuracy on another subset.
- Analyze and report the results

Data set

- The data contains 101 classes, with 31-800 images each.
- We loaded the images, got them to gray scale, and resized them according to the hyper parameter S.
- Data split:
    - We debugged and tuned the pipe on the first 10 classes (fold 1).
    - We ran the algorithm with the best hyper parameter configuration found, on classes 11-20 (fold 2).
- We trained and tested it: training on 20 images per class , testing on 20 others (unless there are less than 40 images for the class – in which case we had less images for test)

Implemented pipe

The pipe we chose to implement is SVM + K-Means + SIFT. The SVM classifier was trained with two types of kernel (linear and RBF) as described below.

Hyper Parameters Tuning

Parameters:

- **Image Subset** – Percentage of images taken from each class for training the dictionary with K-means model = 0.7
- **SIFT Subset** - Percentage from each image's SIFTs taken for training the dictionary with K-means model = 0.1
  In this project we had some time constraint, as a result, we chose these values instead of taking the biggest subsets' sizes that may improve the accuracy.

Manually tuned:

Manually Parameter	Description	Possible Values
S	the image sizes	100, 150, 200
K	the number of codewords for K-means	100, 250, 500
Step size	the number of pixels between each key point	5, 7, 10
Scale	the effect radius over the key point	[4,8,12,16], [5,10,15,20]

The tuning of the manual parameters was performed by examining all possible combinations. For each possible combination, we refer to it as fixed parameters and tune the following parameters systematically as described below.

Systematically tuned:

Systematically Parameter	Description	Range	Step
C	the SVM tradeoff parameter-range	(1,100)	5
Kernel	the kernel to use in the SVM	Linear/RBF/Poly	-
Gamma	Kernel coefficient for ‘rbf’, ‘poly’	0.01,0.1,1,5,10,15,20,25,30	-
Degree	the degree of the poly kernel	2,3	

We trained each kernel type separately as follows. First, we trained the linear kernel and tune the c parameter, choosing the c which minimizes the validation error. Second, we trained the RBF kernel, tune the c and gamma parameters, starting with the chosen c in the linear kernel, and loop until convergences in the validation error. Finally, we trained the poly kernel, tuning the c, gamma and degree parameters, starting with the chosen c and gamma in the RBF kernel, and looped until convergence in the validation error. The chosen systematically tuned parameters (for a specific combination of fixed manually parameters) are those with the lowest validation error.

Validation Results

<u>Manually tuned parameters</u>:

Manually Parameters				Validation error
image size	K	Step size	scale	
150	250	10	[4,8,12,16]	32.05%
150	250	5	[4,8,12,16]	32.5%
150	500	10	[5,10,15,20]	30%
150	500	5	[5,10,15,20]	32.5%
150	500	7	[5,10,15,20]	22.5%
200	500	10	[4,8,12,16]	30%
200	500	10	[5,10,15,20]	30%
200	500	5	[5,10,15,20]	27.5%
200	250	10	[5,10,15,20]	40%
200	500	7	[5,10,15,20]	30%
100	500	5	[4,8,12,16]	30%
100	500	10	[5,10,15,20]	32.5%

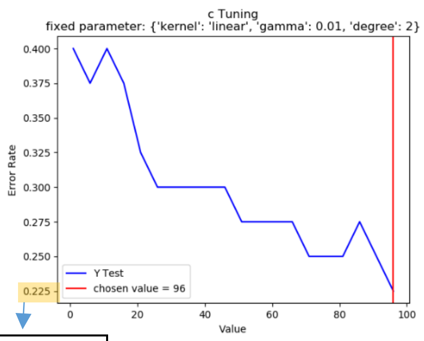
100	500	7	[4,8,12,16]	32.5%
100	500	7	[5,10,15,20]	30%
100	100	5	[5,10,15,20]	37.5%

Systematically tuned parameters:

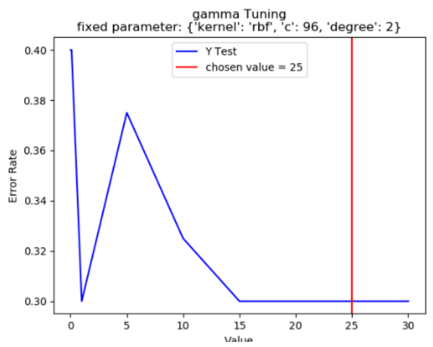
We chose to present only the graphs of the best validation error we got from the manually tuning.

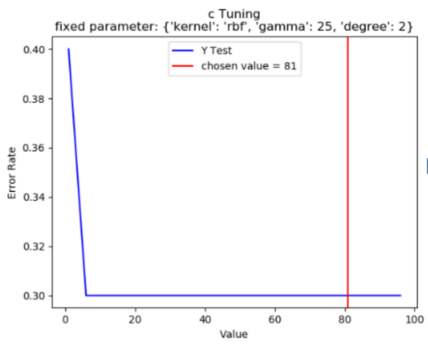
(we got several graphs for each manually tuning parameters combination)

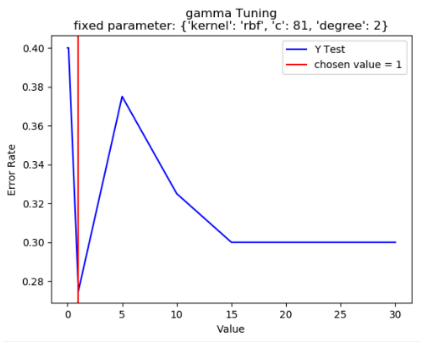
image size = 150, K = 500, step size = 7, scale = [5, 10, 15, 20], {'c': 96, 'kernel': 'linear', 'gamma': 25, 'degree': 2}:

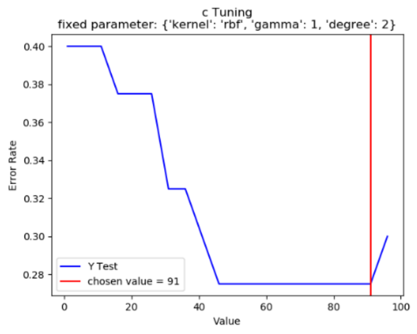


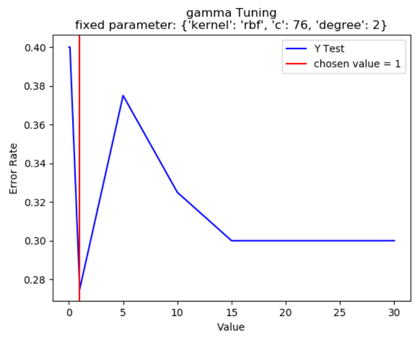
Minimum error

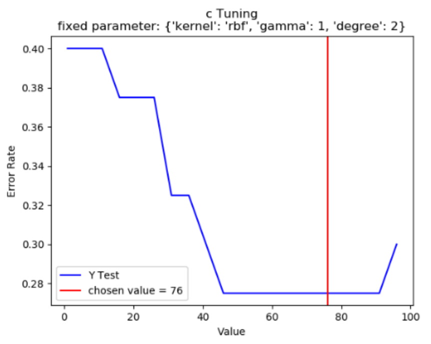


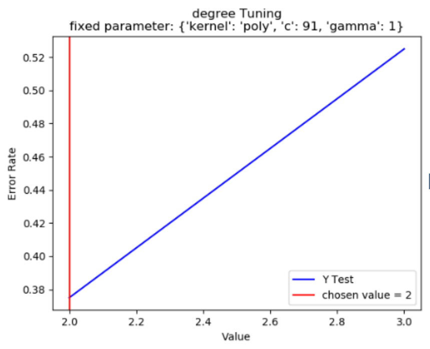


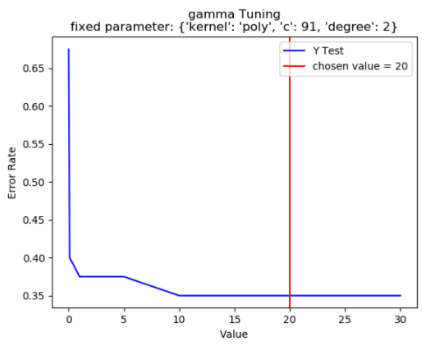


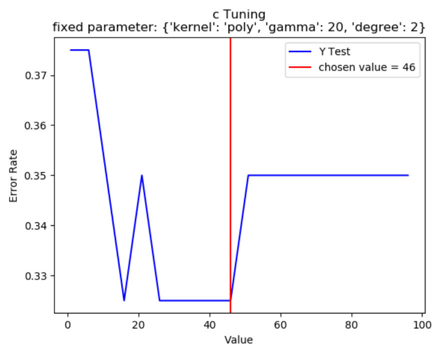


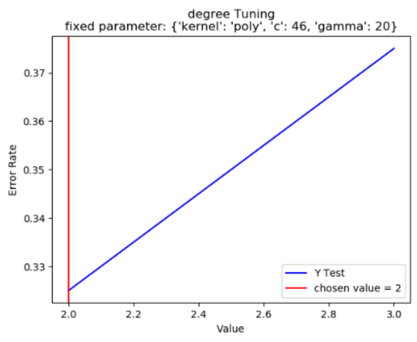


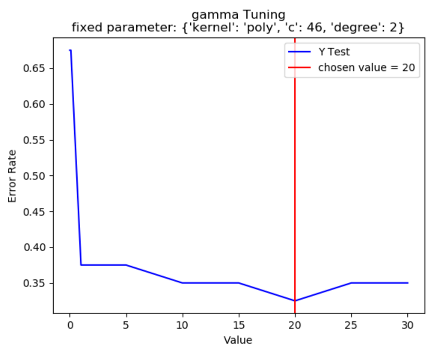


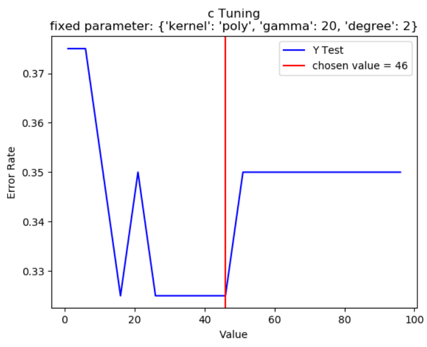










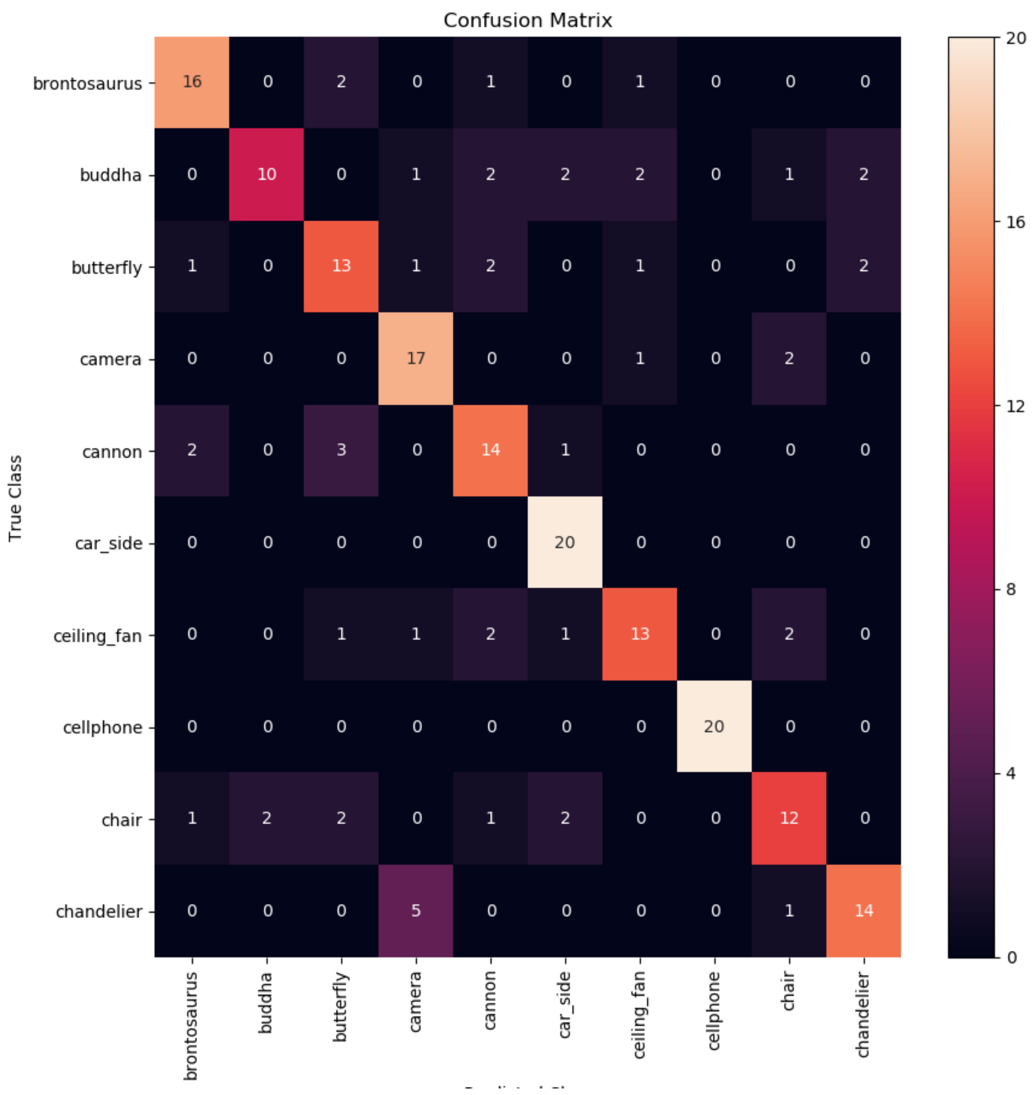


Test Results

Error rate:

The error rate obtained over the test set is: **25.5%**

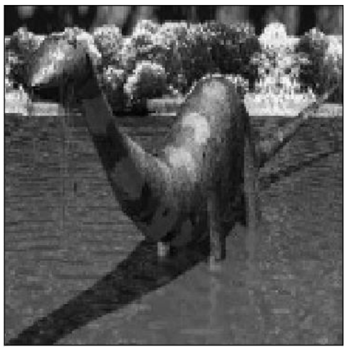
Confusion Matrix:



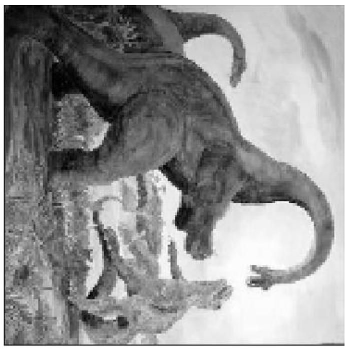
True Class	brontosaurus	buddha	butterfly	camera	cannon	car_side	ceiling_fan	cellphone	chair	chandelier
brontosaurus	16	0	2	0	1	0	1	0	0	0
buddha	0	10	0	1	2	2	2	0	1	2
butterfly	1	0	13	1	2	0	1	0	0	2
camera	0	0	0	17	0	0	1	0	2	0
cannon	2	0	3	0	14	1	0	0	0	0
car_side	0	0	0	0	0	20	0	0	0	0
ceiling_fan	0	0	1	1	2	1	13	0	2	0
cellphone	0	0	0	0	0	0	0	20	0	0
chair	1	2	2	0	1	2	0	0	12	0
chandelier	0	0	0	5	0	0	0	0	1	14

Largest Errors:

Brontosaurus

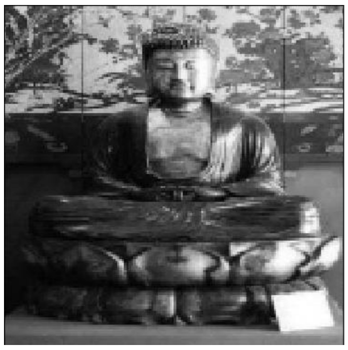


Error: 0.289

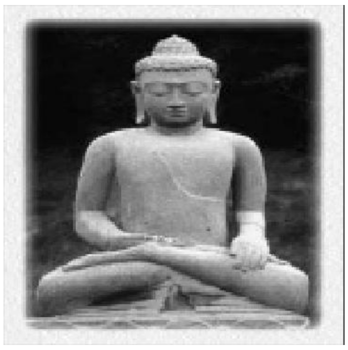


Error: 1.47

Buddha

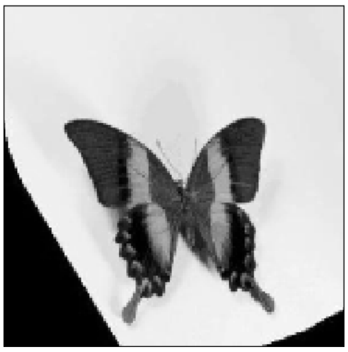


Error: 0.80

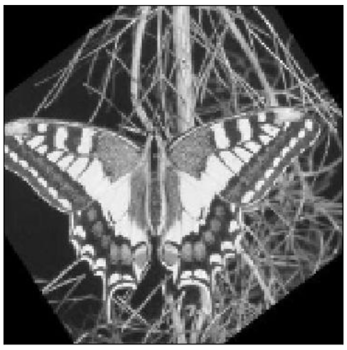


Error: 0.924

Butterfly



Error: 0.416



Error: 0.745

Camera

Error: 0.764

Error: 1.042

Cannon

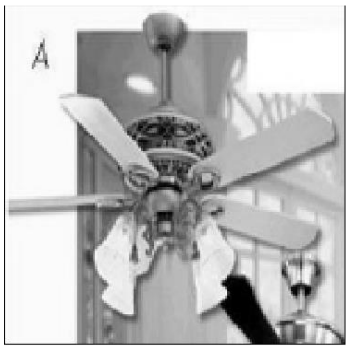
Error: 1.317

Error: 1.172

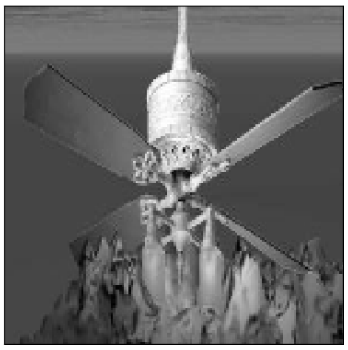
Car_side

There were no errors for this class

Ceiling_fan



Error: 0.467



Error: 0.956

Cellphone

There were no errors for this class

Chair

Error: 0.539

Error: 0.646

Chandelier

Error: 0.824

Error: 1.396