Image Classification

Cheick Berthe, Joshua Knestaut, Matthew Stevens

Cifar-10 Dataset: 50,000 data points

Airplane

Automobile

Truck

Ship

Bird

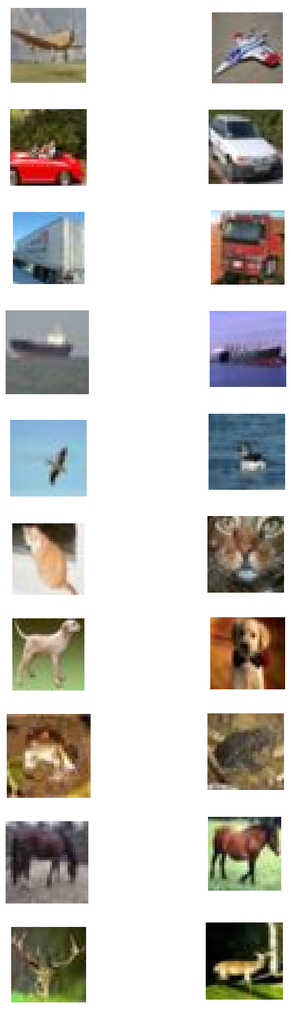
Cat

Dog

Frog

Horse

Deer



Introduction

This project was largely inspired by a collective want of better results for the problem presented in the first homework of ECE 4424. In that homework, the task was to classify images of outdoor scenes using the KNN algorithm. This project models itself after that homework by using the same feature extraction method, but attempts to improve the accuracy while using a larger number of classes.

Approach

Originally, sights were set high trying to classify an objects within a superclass and the subclass (as presented in the CIFAR-100 dataset), until it was realized how difficult it is to classify objects visually. It was decided that the standard learning algorithms should be compared based on how accurately they can classify the data. We chose to use variations of KNN, Naive Bayes and SVM to classify the CIFAR-10 dataset.

Discussion

It is obvious in 2016 that the best method for object classification would be a convolutional neural network. With the skill, experience, and time available, that route was abandoned in favor of optimizing simpler algorithms.
We also discovered that with increased data and classes, training and classifying the data takes significant time and computational power. It is now understood why supercomputers or large scale computer networks are used to solve large machine learning problems.

Conclusions

Accuracy was overall lower than expected due to a high class coun and many similarities between the classes. The best classifier was a bagged classifier composed of a series of SVM's with both RBF and quadratic kernels trained on different sets of data, and the 20-Nearest Neighbors classifier with euclidean distance. Some noteworthy results:

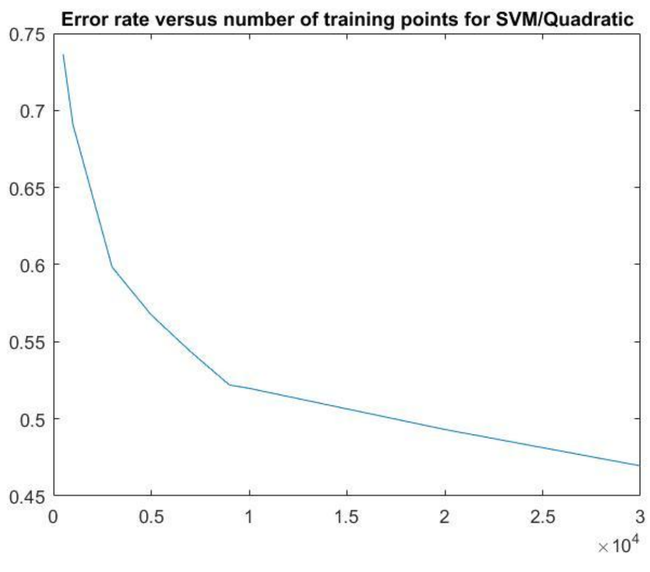
- Cats were often misclassified as trucks
- Dogs mostly misclassified as cats
- Large percentage of cats were being misclassified

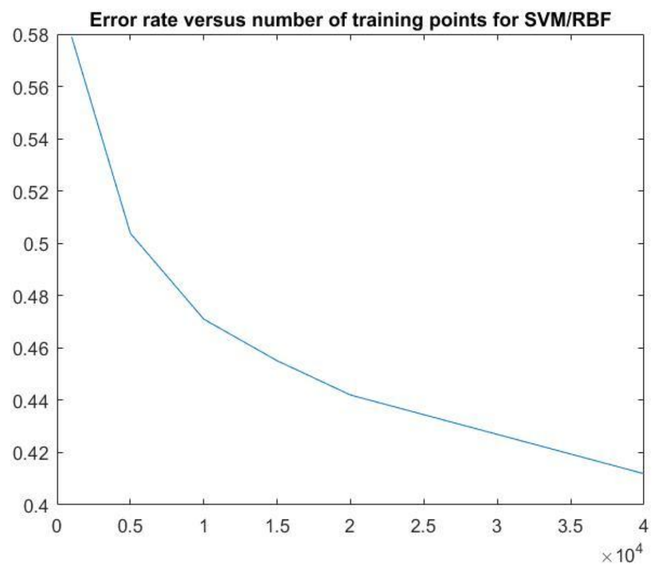
More information, results, and algorithms used can be found at github.com/kjosh9/ML_Project

Feature Extraction

SIFT and SURF feature extraction was contemplated, but they do not produce fixed length vectors, therefore they would require their own algorithms, separate from GIST.

GIST was selected as the feature extraction method for this project due to its simplicity and consistent size of produced data.





Support Vector Machine

Methods:

- Used an adapted version of SVM to work with multiple classes
- Also created a bagged version to increase the accuracy.
- Compared kernels (rbf, linear, quadratic, etc.)

Results and Observations:

- Excessive computation time
- With the large amount of data, requires a large number of iterations to converge (>=1,000,000)
- Requires a large amount of training data to be more accurate
- A sigma value of 16 for the RBF kernel produced the least error
- Bagged SVM produced the least error (61% correct)

K-Nearest Neighbors

Methods:

- Used as a baseline due to the simplicity of the algorithm.
- Compared different distance metrics (euclidean, Manhattan/city, Mahalanobis, etc.)
- Optimized over K

Results and Observations:

- Distance metric made little change in the classifier's accuracy.
- Surprisingly accurate with a relatively small amount of data
- Mahalanobis distance took too long to train on significant amount of data.
- Ran into the curse of dimensionality with KNN

Multinomial Gaussian Naive Bayes

Assumptions:

- Conditionally independence
- Uniform priors

$$\begin{bmatrix} (\mu_{1,1},\hat{\sigma}_{1,1}) & & (\mu_{1,f},\hat{\sigma}_{1,f}) \\ & (\mu_{i,j},\hat{\sigma}_{i,j}) & \\ (\mu_{c,1},\hat{\sigma}_{c,1}) & & (\mu_{c,f},\hat{\sigma}_{c,f}) \end{bmatrix}$$

Learning algorithm:

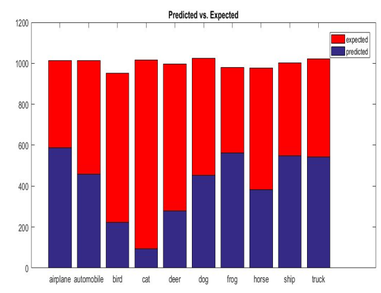
```
for each class c
  for each feature f that maps to c
    fit Gaussian to feature values
    Store in distribution matrix
at(c,f)
```

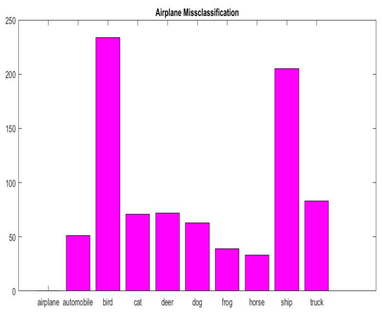
Prediction:

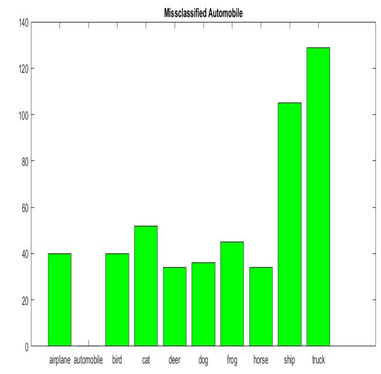
```
for each class c
  Store sum of log pdf for each feature
Choose class with maximum probability
```

Results and Observations:

- Higher accuracy than expected
- Lower accuracy overall due to violations of the conditional independence assumption







Future Work

Future work on this topic will include creating a CNN to increase accuracy (though several extremely accurate networks exist currently). Implementations using SIFT and SURF features will also be created, though they will require their own non-generic algorithms.