

# Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines

JINHO KIM<sup>1</sup>

Okemos High School

2800 Jolly Road

Okemos, MI 48864

<sup>1</sup>jcykim2008@gmail.comBYUNG-SOO KIM<sup>2</sup>, SILVIO SAVARESE<sup>3</sup>

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109-2122

<sup>2</sup>bsookim@umich.edu, <sup>3</sup>silvio@eecs.umich.edu

**Abstract:** - In order for a robot or a computer to perform tasks, it must recognize what it is looking at. Given an image a computer must be able to classify what the image represents. While this is a fairly simple task for humans, it is not an easy task for computers. Computers must go through a series of steps in order to classify a single image. In this paper, we used a general Bag of Words model in order to compare two different classification methods. Both K-Nearest-Neighbor (KNN) and Support-Vector-Machine (SVM) classification are well known and widely used. We were able to observe that the SVM classifier outperformed the KNN classifier. For future work, we hope to use more categories for the objects and to use more sophisticated classifiers.

**Key-Words:** - Bag of Words Model, SIFT (Scale Invariant Feature Transform), *k*-means clustering, KNN (k-nearest-neighbor), SVM (Support Vector Machine)

## 1 Introduction

The human ability to analyze and classify objects and scenes rapidly and accurately is something that everybody finds highly useful in everyday. Thorpe and his colleagues found that humans are able to categorize complex natural scenes very quickly [1]. In order to understand a complex scene, the first step is to recognize the objects and then recognize the category of the scene [2]. In order to do this in computer vision, we use various classifiers that all have different characteristics and features.

In the past, many classifiers have been developed by various researchers. These methods include naïve Bayes classifier, support vector machines, k-nearest neighbors, Gaussian mixture model, decision tree and radial basis function (RBF) classifiers [3,4]. These classifiers are used in algorithms that involve object recognition. However object recognition is challenging for several reasons. The first and most obvious reason is that there are about 10,000 to 30,000 different object categories. The second reason is the viewpoint variation where many objects can look different from different angles. The third reason is illumination in which lighting makes the same objects look like different objects. The fourth reason is background clutter in which the classifier cannot distinguish the object from its background. Other challenges include scale,

deformation, occlusion, and intra-class variation. Applications for classification in computer vision include computational photography, security, surveillance, and assistive driving.

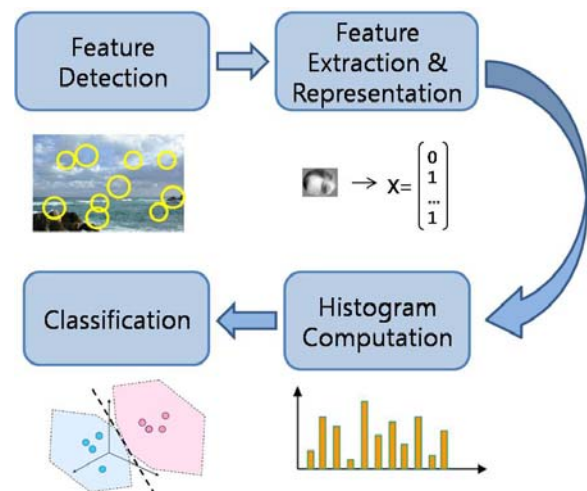


Fig. 1 A conceptual illustration of the process of image classification.

A typical classification method using the bag of words model consists of four steps as shown in Fig.1 In short, the bag of words model creates histograms of images which is used for classification.

In this paper, we will be comparing two different classification methods: Experimental evaluation is conducted on the Caltech-4-Cropped dataset [5] to see the difference between two classification methods. In Section 2, we will discuss and outline our bag of words Model. In Section 3, we will explain the two different classification methods we have used: KNN and SVM.

## 2 Image Representation - Bag of Words Model

One of the most general and frequently used algorithms for category recognition is the bag of words (abbreviation BoW) also known as bag of features or bag of keypoints model [6, 7]. This algorithm generates a histogram, which is the distribution of visual words found in the test image, and then classifiers classify the image based on each classifier's characteristics. The KNN classifier compares this histogram to those already generated from the training images. In contrast, the SVM classifier uses the histogram from a test image and a learned model from the training set to predict a class for the test image. The purpose of the BoW model is representation. Representation deals with feature detection and image representation. Features must be extracted from images in order to represent the images as histograms.

Section 2.1 deals with the feature extraction process of the BoW model. We extracted features using SIFT [8]. Section 2.2 deals with clustering the features extracted in Section 2.1 by  $k$ -means clustering. Section 2.3 deals with the histogram computation process.

### 2.1 Scale-Invariant Feature Transform

The first step for our two classification methods is to extract various features the computer can see in an image. For any object in an image, there are certain features or characteristics that can be extracted and define what the image is. Features are then detected and each image is represented in different patches. In order to represent these patches as numerical vectors we used SIFT descriptors to convert each patch into a 128-dimensional vector. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination, which is why we used SIFT descriptors. After converting each patch into numerical vectors, each image is a collection of

128-dimensional vectors. Fig. 2 shows SIFT descriptors in work on one of our images of airplanes. The SIFT algorithm we used was from the VLFEAT library [9], which is an open source library for popular computer vision algorithms.

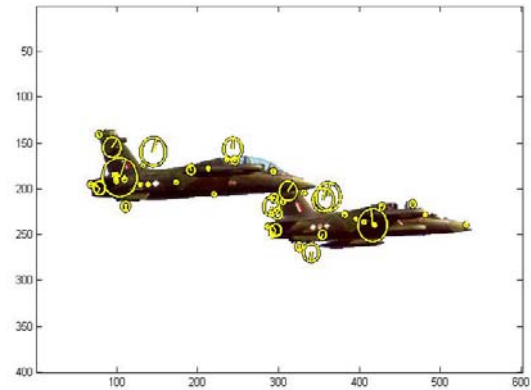


Fig. 2 Features extracted from an image of airplanes by using SIFT. The circles represent the various features detected by the SIFT descriptor.

### 2.2 $k$ -means Clustering

After extracting features from both testing and training images, we converted vector represented patches into codewords. To do this we performed  $k$ -means clustering over all the vectors.  $k$ -means clustering is a method to cluster or divide  $n$  observations or, in our case, features into  $k$  clusters in which each feature belongs to the cluster of its nearest mean [10]. We cluster our features and prepare the data for histogram generation. As the number of clusters is  $k$ , an input, an inappropriate choice of  $k$  may yield poor results. Therefore to prevent this problem, we tested classification with 5 different  $k$  values or codewords: 50, 100, 250, 500, and 1000. These codewords are defined as the centers of each cluster. The number of codewords is the size of each codebook.

### 2.3 Histogram Generation

Each patch in an image is mapped to a certain codeword through the  $k$ -means clustering process and thus, each image can be represented by a histogram of the codewords. This is the final step before the actual classification, which is to generate histograms of the features extracted in each image [11]. These features are stacked according to which cluster they were clustered in by  $k$ -means clustering.

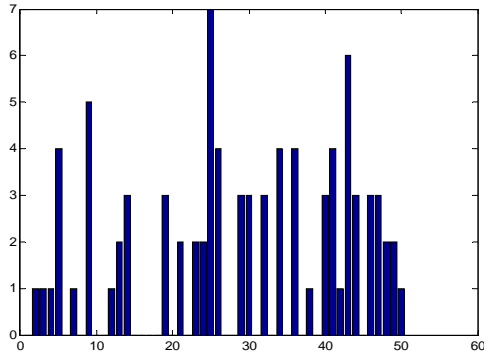


Fig.3 Histogram generated from image of airplanes from Fig. 2

### 3 Methods/Results

Once the BoW model represents the images as certain features, the next step towards classification is learning. Learning is what is referred to as the “training process” where the classifier learns different features for different categories and forms a codeword dictionary. It is essential to have a solid training process which is why the train ratio is greater than the test ratio. The final step is classification or recognition during which the classifier tests or classifies an image based on the similarities between the feature extracted and the codeword dictionary produced through the training process of the algorithm.

Section 3.1 deals with the two classification methods used in this research: KNN and SVM. Section 3.2 shows the classification results of the two classifiers.

#### 3.1 Classification Methods

The problem of object classification can be specified as a problem to identify the category or class that the new observations belong to based on a training dataset containing observations whose category or class is known.

The purpose of the BoW Model was for image representation. As shown in Sections 2.1 and 2.2, the features detected by SIFT descriptors were put into a codebook by  $k$ -means clustering. Classification can now be performed by comparing histograms representing the codewords.

Generally, classification works by first plotting training data into multidimensional space. Then each classifier plots testing data into the same multidimensional space as the training data and compares the data points between testing and training to determine the correct class for each

individual query point. In Section 3.1.1, we will discuss KNN classification while in Section 3.1.2, we will discuss SVM classification.

##### 3.1.1 K-Nearest-Neighbor Classification

$k$ -nearest neighbor algorithm [12,13] is a method for classifying objects based on closest training examples in the feature space.  $k$ -nearest neighbor algorithm is among the simplest of all machine learning algorithms. Training process for this algorithm only consists of storing feature vectors and labels of the training images. In the classification process, the unlabelled query point is simply assigned to the label of its  $k$  nearest neighbors.

Typically the object is classified based on the labels of its  $k$  nearest neighbors by majority vote. If  $k=1$ , the object is simply classified as the class of the object nearest to it. When there are only two classes,  $k$  must be a odd integer. However, there can still be ties when  $k$  is an odd integer when performing multiclass classification. After we convert each image to a vector of fixed-length with real numbers, we used the most common distance function for KNN which is Euclidean distance:

$$\begin{aligned} d(x, y) &= \|x - y\| = \sqrt{(x - y) \cdot (x - y)} \\ &= (\sum_{i=1}^m ((x_i - y_i)^2))^{1/2} \end{aligned} \quad (1)$$

where  $x$  and  $y$  are histograms in  $X = R^m$ . Fig. 4 shows visualizes the process of KNN classification.

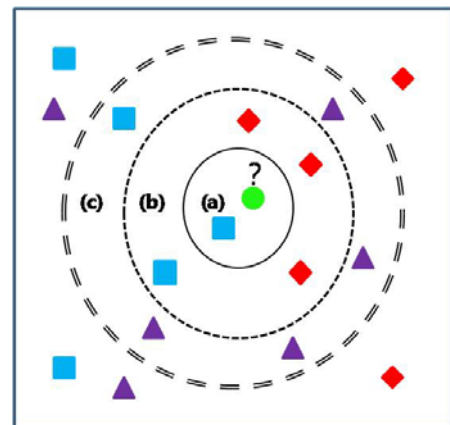


Fig. 4 KNN Classification. At the query point of the circle depending on the  $k$  value of 1, 5, or 10, the query point can be a rectangle at (a), a diamond at (b), and a triangle at (c).

A main advantage of the KNN algorithm is that it performs well with multi-modal<sup>2</sup> classes because the basis of its decision is based on a small neighborhood of similar objects. Therefore, even if the target class is multi-modal, the algorithm can still lead to good accuracy. However a major disadvantage of the KNN algorithm is that it uses all the features equally in computing for similarities. This can lead to classification errors, especially when there is only a small subset of features that are useful for classification.

### 3.1.2 Support Vector Machine Classification

SVM classification [14] uses different planes in space to divide data points using planes. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories or classes are divided by a dividing plane that maximizes the margin between different classes. This is due to the fact if the separating plane has the largest distance to the nearest training data points of any class, it lowers the generalization error of the overall classifier. The test points or query points are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on as shown in Figure 5.

Since MATLAB's SVM classifier does not support multiclass classification and only supports binary classification, we used lib-svm toolbox, which is a popular toolbox for SVM classification. Since the cost value, which is the balancing parameter between training errors and margins, can affect the overall performance of the classifier, we tested different cost parameter values among 0.01~10000 to see which gives the best performance using validation set.

As mentioned before, the goal of SVM Classification is to produce a model, based on the training data, which will be able to predict class labels of the test data accurately. Given a training set of instance-label pairs  $(x_i, y_i), i = 1, \dots, l$  where  $x_i$  and  $y_i$  are histograms from training images,  $x_i \in R^n$  and  $y \in \{1, -1\}^l$ , [15,16] the support vector machines can be found as the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2)$$

$$\text{Subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

<sup>2</sup> Multi-modal: consisting of objects whose independent variables have different characteristics for different subsets

The function  $\phi$  maps training vectors  $x_i$  into a higher dimensional space while  $C > 0$  is the penalty parameter for the error term [17]. We used two different basic kernels: linear and radial basis function (RBF). The linear kernel function and the radial basis function can be represented as,

$$K(x_i, x_j) = x_i^T x_j \quad (3)$$

$$\text{and} \\ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (4)$$

respectively, while  $\gamma$  is a kernel parameter. Fig. 5 shows a visualization of the process of SVM classification.

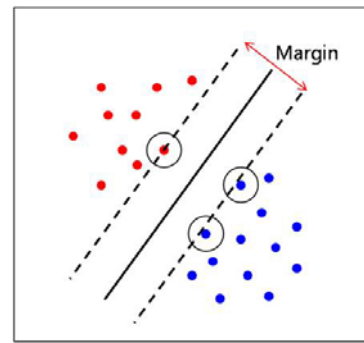


Fig. 5 SVM Classification. In multidimensional space, support vector machines find the hyperplane that maximizes the margin between two different classes. Here the support vectors are the dots circled.

A main advantage of SVM classification is that SVM performs well on datasets that have many attributes, even when there are only a few cases that are available for the training process. However, several disadvantages of SVM classification include limitations in speed and size during both training and testing phase of the algorithm and the selection of the kernel function parameters.

## 3.2 Results

We performed classification with a 5-fold validation set, each fold with approximately 2800 training images and approximately 700 testing images. Each experiment had different images in training and testing compared to the other experiments so as to prevent the overlapping of testing and training images in each experiment.

In order to compare the performance levels of KNN classification and SVM classification, we have implemented the framework for the BoW model as discussed in section 2.





Fig. 6 Sample test images of the 4 classes. Starting on the upper right and in clockwise order: Cars, Motorbikes, Faces, Airplanes.

We have mainly used MATLAB as our computer language but have reverted back and forth occasionally with the windows command prompt in order to run multiclass classification with SVM. Our dataset, which contains 3505 images from 4 different classes (airplanes, cars, faces, motorbikes), was divided into testing images and training image by a 1:5 ratio in order to increase diversity within each experiment. The results are mainly shown in the form of confusion tables (or confusion matrix) which are useful for comparison of classifiers performance among the different classes. We discuss KNN classification results in Section 3.2.1 and we discuss SVM classification in Section 3.2.2.

### 3.2.1 KNN Classification Results

The best number of codewords for KNN classification was 50 codewords. This was mainly due to the fact that our algorithm could not classify cars as well as the other classes. In general, once we moved on to other number of codewords apart from 50, the accuracy of the overall classifier dropped dramatically due to the algorithm's inability to classify cars. For the other codewords, the classification accuracy for the other classes rose but did not rise enough to compensate for the lost of accuracy for cars. We used Euclidean distance as the distance metric and since the constant value of  $k$  can influence the accuracy of the overall classification, we tested with 5 different  $k$  values: 1, 5, 10, 15, and 20. Overall the best value for the constant  $k$  was 5. Table 1 shows the average performance of the classifier for each individual class with the actual class represented on the left and the predicted class represented above. The diagonally shaded boxes show the percent of accurately classified images for each class while the other boxes show the percent of erroneously classified images. The average accuracy for the overall classifier with the best  $k$  value was 78.03%.  $k$ -nearest neighbor algorithm

Table 1. The average performance of KNN classifier

	Airplanes	Cars	Faces	Motorbikes
Airplanes	87.24%	0.09%	8.47%	4.19%
Cars	5.28%	61.13%	7.79%	25.80%
Faces	5.57%	0%	87.74%	6.69%
Motorbikes	5.56%	0%	10.05%	84.39%

### 3.2.2 SVM Classification Results

The number of codewords to produce the best classification results for SVM turned out to be 500 codewords. The overall best cost value for SVM was 10000. The overall accuracy rate for the SVM classifier was 92%. We found that unlike KNN classification, SVM classification was much more adept at classifying cars which is the main reason why the accuracy rate for SVM classification was much better compared to KNN classification. Table 2 shows the average performance of the SVM classifier for each individual class with the actual class represented on the left and the predicted class represented above. Like Table 1, the diagonally shaded boxes show the percent of accurately classified images for each class while the other boxes show the percent of erroneously classified images. The average accuracy for the overall classifier with the best cost value of 10000 was 91.9%.

Table 2. The average performance of SVM classifier

	Airplanes	Cars	Faces	Motorbikes
Airplanes	93.39%	0.65%	3.45%	2.51%
Cars	0.78%	96.79%	0.35%	2.08%
Faces	4.46%	1.34%	86.85%	7.36%
Motorbikes	3.03%	4.36%	2.06%	90.56%

## 4 Conclusion

In this work, we implemented two different frameworks for image classification. With a proper learning process, we could observe that SVM classification outperformed KNN classification. Although the performance of KNN was very low compared to SVM, this was due to the fact that the KNN classifier could not distinguish cars whereas SVM was very effective in classifying cars. If we take the class of cars away from the overall experiment we can see that SVM only outperforms KNN by 4%. If we add the cars back into the overall experiment however, we can see that SVM

outperforms KNN by 12%. To improve upon this work, we can extract individual features from the cars in our Caltech-4-Cropped dataset and investigate why SVM was so effective in classifying cars compared to KNN.

The performance of various classification methods still depend greatly on the general characteristics of the data to be classified. The exact relationship between the data to be classified and the performance of various classification methods still remains to be discovered. Thus far, there has been no classification method that works best on any given problem. There have been various problems to the current classification methods we use today. To determine the best classification method for a certain dataset we still use trial and error to find the best performance. For future work, we can use more different kinds of categories that would be difficult for the computer to classify and compare more sophisticated classifiers. Another area for research would be to find certain characteristics in various image categories that make one classification method better than another.

### Acknowledgements

We would like to thank Mr. Andrew Moore, Research Instructor of Okemos High School Research Seminar Program, for creating this opportunity for research.

### References:

- [1] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381, 1996, pp. 520–522.
- [2] A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12, 1980, pp. 97–136.
- [3] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011.
- [4] D.A. Forsyth and J. Ponce, *Computer Vision, A Modern Approach*, Prentice Hall, 2003.
- [5] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *IEEE Computer Vision Pattern Recognition 2004*, Workshop on Generative-Model Based Vision. 2004.
- [6] Lewis, David, Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval, *Proceedings of ECML-98*, 10th European Conference on Machine Learning. Springer Verlag, 1998, pp. 4–15.
- [7] S. Savarese, J. Winn, and A. Criminisi, Discriminative Object Class Models of Appearance and Shape by Correlations, *Proc. of IEEE Computer Vision and Pattern Recognition*. 2006.
- [8] Lowe, David G, Object recognition from local scale-invariant features, *Proceedings of the International Conference on Computer Vision*. Vol. 2. 1999, pp. 1150–1157.
- [9] A. Vedaldi and B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, <http://www.vlfeat.org/>, 2008
- [10] Chris Ding and Xiaofeng He, K-means Clustering via Principal Component Analysis, *Proceeding of International Conference Machine Learning*, July 2004, pp. 225-232.
- [11] Rafael G. Gonzales and Richard E. Woods, *Digital Image Processing*, Addison Wesley, 1992.
- [12] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G, Output-sensitive algorithms for computing nearest-neighbor decision boundaries, *Discrete and Computational Geometry*, 2005, pp. 593–604.
- [13] Cover, T., Hart, P. , Nearest-neighbor pattern classification, *Information Theory, IEEE Transactions on*, Jan. 1967, pp. 21-27.
- [14] Christopher J.C. Burgers, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 1998, pp. 121–167.
- [15] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM Press, 1992, pp. 144-152.
- [16] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20, 1995, pp. 273-297.
- [17] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification (Technical report). Department of Computer Science and Information Engineering, National Taiwan University.