**Object Classification**

**By**: Chandni Shankar (cxs152830)

Contents

[1 Introduction 3](#_Toc449825713)

[2 Approach 3](#_Toc449825714)

[2.1 Feature Extraction 4](#_Toc449825715)

[2.1.1 SIFT 4](#_Toc449825717)

[2.1.2 Bag of Features method 5](#_Toc449825719)

[2.2 Training and Classification 6](#_Toc449825721)

[2.2.1 Classifiers 6](#_Toc449825722)

[3 Experiment and Results 6](#_Toc449825723)

[3.1 Keypoints Extracted for Images 6](#_Toc449825724)

[3.2 Experiment Results 7](#_Toc449825725)

[3.3 Correct Classifications 11](#_Toc449825729)

[3.4 Misclassifications 12](#_Toc449825730)

[4 Future Work 12](#_Toc449825731)

[5 Conclusions 12](#_Toc449825732)

[6 References 12](#_Toc449825733)

# Introduction

Object detection and classification has become essential in our daily activities as a multitude of important applications including face detection for security or object detection for automatic traffic maneuvering involve detecting and categorizing object. The aim of this paper is to perform multi-class object categorization using different machine learning algorithms and compare the performance of the classifiers on the task.

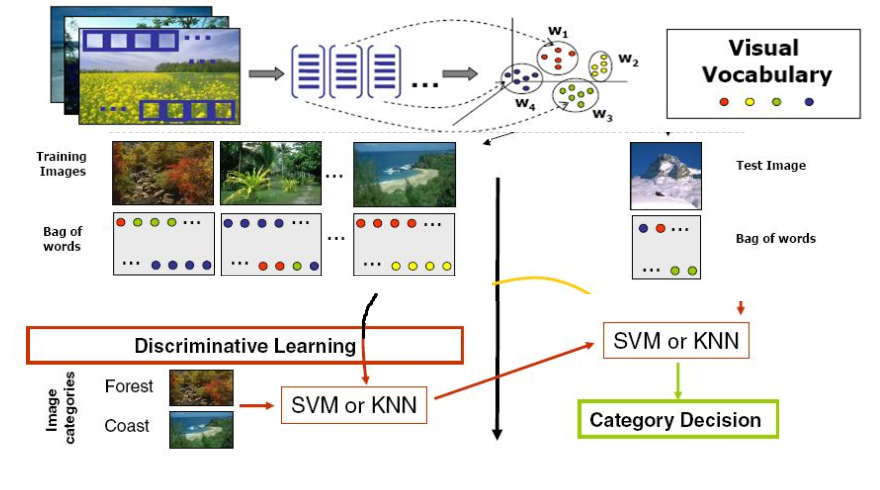
The images are represented by a matrix of their pixel values (bitmap representation) in different backgrounds, orientation, scale and illumination. The proposed system performs SIFT feature extraction to robustly identify the key-points describing the image and uses a popular computer vision concept of bag-of-features to represent a histogram of these features to train different classifiers to categorize the objects. The recognition rates achieved in our experiments indicate that Random Forest are well-suited for recognition of multi-class objects.

# Approach

Our project uses the Caltech 256 dataset to illustrate the multi-class performed by different classifiers namely Linear SVM, Radial Basis Function SVM, K Nearest Neighbor and Random Forest for a subset of the data. The images are represented by a matrix of their pixel values after converting to gray scale to reduce the size and time for processing. The dataset is split into training and test datasets using a 4:1 split to validate our classifiers.

## Feature Extraction

## The objects in the images are present at different orientation, illumination, occlusions, scale and backgrounds. The recognition of objects is therefore not an easy task at hand. In order to help us with efficient classification, features that describe the image best is extracted from each category of image using the Scale Invariant Feature Transform (SIFT) technique.

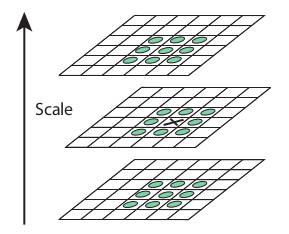


### SIFT

### Among the different techniques available for feature extraction in an image, SIFT proves very efficient. This feature extraction proposed by David Lowe identifies the key points that distinguish the image and computes its descriptors to provide information about the key point.

#### Scale -space Extrema Detection

#### The algorithm finds points by performing the Laplacian of Gaussians (LoG) for the image at different scales. LoG acts as a blob detector which detects blobs in various sizes due to change in scale. But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. It identifies the keypoints by finding the maximum wrt to 27 other points across different scales once this DoG are found as the maximum point doesn’t vary with scale



#### Keypoint Localization

#### Once potential keypoints locations are found, they have to be refined to get more accurate results. It eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

#### Orientation Assignment

#### Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neigbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

#### Keypoint Descriptor

#### Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is devided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor.

#### Keypoint Matching

#### Keypoints between two images are matched by identifying their nearest neighbours.

### Bag of Features method

### Similar to the Bag of Words concept with text classification, Bag of Features uses a sparse histogram representation of different patches of the images. The vocabulary is formed using feature detection, feature description and a codebook generation.

#### Feature Description

#### This stage of the Bag of Features model is performed by applying the SIFT algorithm to each image to extract features that best describe the image invariant to light intensity, rotation, orientation and affine variations to some extent.

#### Codebook Generation

#### A code word is a representation of similar patches of images. This is obtained by performing the K means clustering algorithm to find the mean centroids of the image patches and number of clusters forms the code words and visual vocabulary.

## Training and Classification

Classification of the images is done by training a classifier using useful machine learning algorithms. We ran the training for SVM (Linear & RBF Kernel), K Nearest Neighbors and Random forest. The classifiers each use the feature values to calculate associated weights of each feature value which is used to predict the test image.

### Classifiers

As part of Image classification we have used following algorithms.

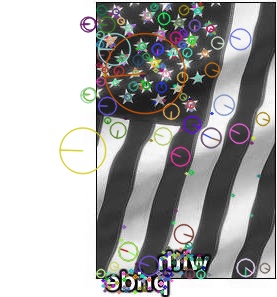
1. Linear SVM Algorithm
2. RBF Kernel Algorithm
3. KNN Algorithm
4. Random Forest Algorithm

# Experiment and Results

We ran the dataset against 4 classifiers as mentioned earlier for different number of classes and parameter to fine-tune our results. Below are a few plots representing the classification results and accuracy. We have also named each image with the label provided by the classifier to understand which images are misclassified.

## Keypoints Extracted for Images

The circles represent the keypoints identified along with orientation information.



## Experiment Results

Please find below our results with various algorithms and number of classes and respective accuracies

Result 1

|  |  |
| --- | --- |
| # classes | 5 |
| Training images per class | 70 |
| Testing Images per class | 20 |
| Number of clusters | 600 |
| K value in KNN | 15 |

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Linear SVM | 56.1904761905 |
| RBF Kernel | 61.9047619048 |
| KNN | 42.8571428571 |
| RandomForest | 70.4761904762 |

Result 2.

|  |  |
| --- | --- |
| # classes | 10 |
| Training images per class | 70 |
| Testing Images per class | 20 |
| Number of clusters | 600 |
| K value in KNN | 15 |

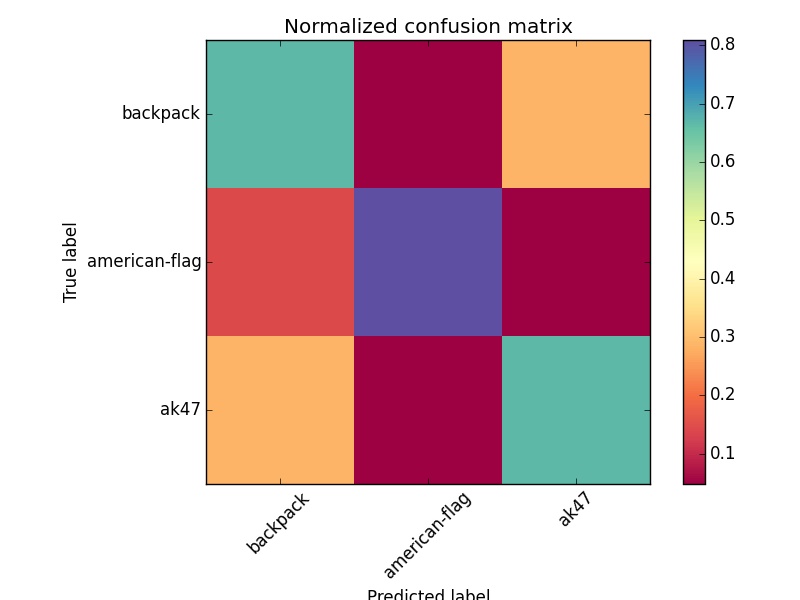
|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Linear SVM | 36.8421052632 |
| RBF Kernel | 48.3253588517 |
| KNN | 29.1866028708 |
| Random Forest | 45.9330143541 |

Result 3.

|  |  |
| --- | --- |
| # classes | 10 |
| Training images per class | 70 |
| Testing Images per class | 20 |
| Number of clusters | 600 |
| K value in KNN | 15 |

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Linear SVM | 44.019138756 |
| RBF Kernel | 43.0622009569 |
| KNN | 25.3588516746 |
| RandomForest | 46.4114832536 |

Linear SVM



Normalized confusion matrix

[[ 0.67 0.05 0.29]

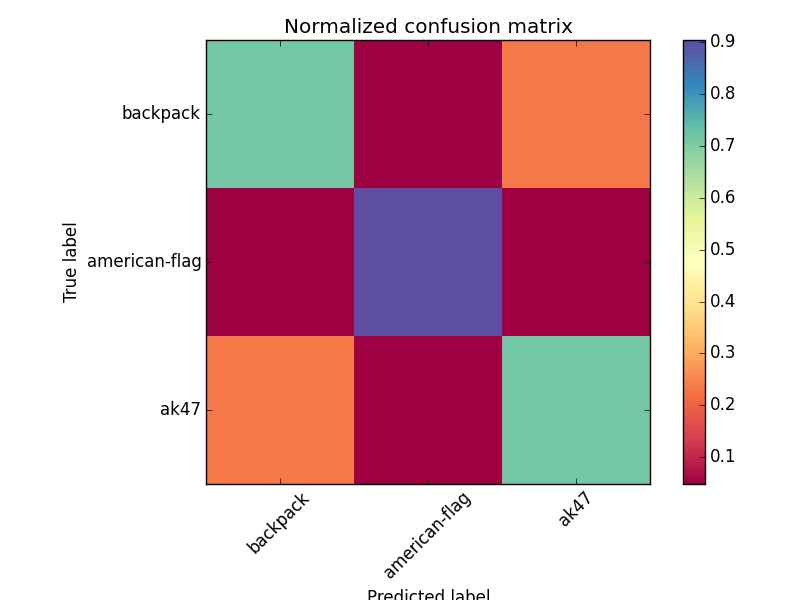
[ 0.14 0.81 0.05]

[ 0.29 0.05 0.67]]

C = 10

Accuracy linearSVM 71.4285714286

## RBF SVM



Normalized confusion matrix

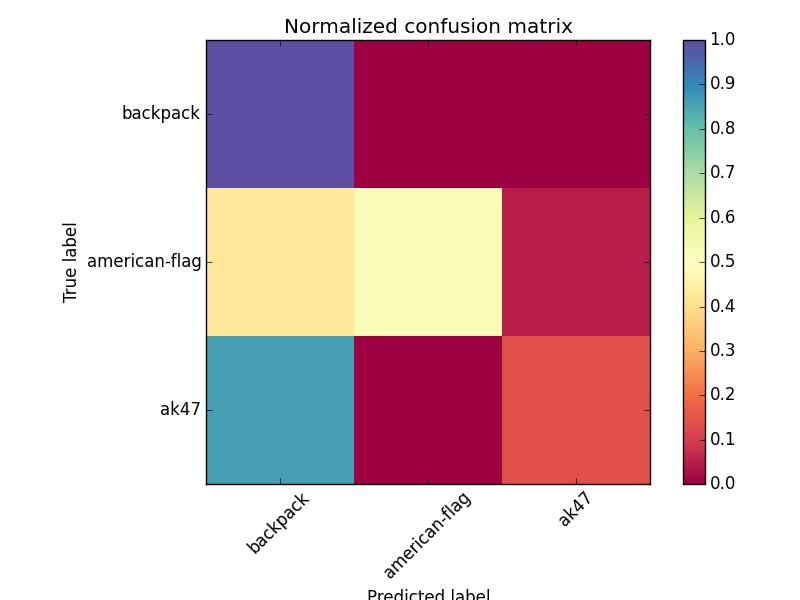
[[ 0.71 0. 0.29]

[ 0.1 0.76 0.14]

[ 0.19 0.05 0.76]]

Accuracy rbfkernel 74.6031746032

## KNN



Normalized confusion matrix

[[ 1. 0. 0. ]

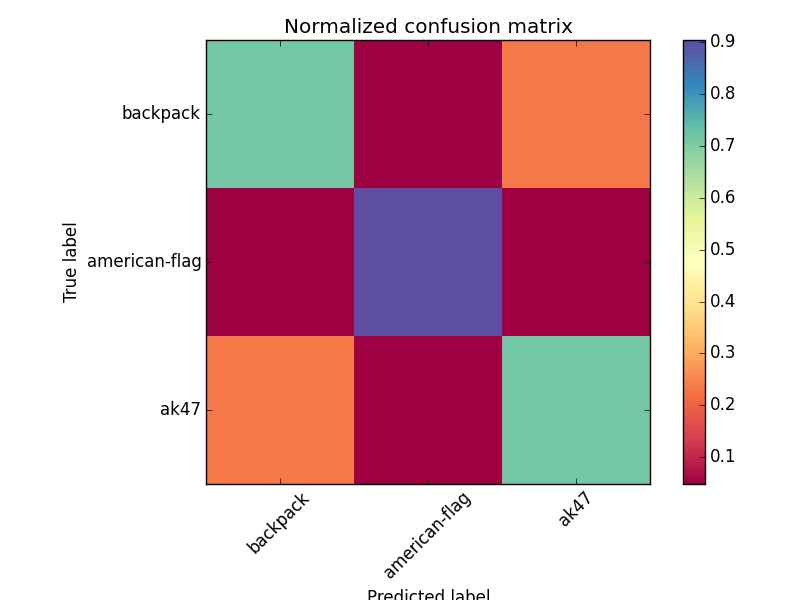
[ 0.43 0.52 0.05]

[ 0.86 0. 0.14]]

No: of nearest neighbors = 10

Accuracy knn 55.5555555556

## Random Forest



Normalized confusion matrix

[[ 0.71 0.05 0.24]

[ 0.05 0.9 0.05]

[ 0.24 0.05 0.71]]

No. of estimators = 100

Accuracy RandomForest 77.7777777778

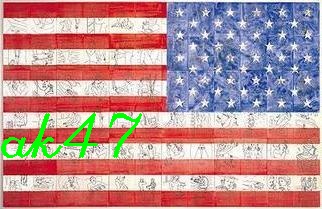
## 

## Correct Classifications



## Misclassifications



# Future Work

We have run the Image classification on the 10 classes with minimum of 80 images for each class for training and 20+ images for testing data. We want to achieve 128 classes in future by changing and improving some areas where we are taking more time.

# Conclusions

We have demonstrated the problem of object classification for multi-class categorization. It appears that object classification with Random forest performs the best among the rest of the classifiers while the least performing classifier seems to be K Nearest Neighbors. This result can however be improved by optimizing calibration parameters and using Deep Neural Networks to retrieve more comprehensive features that would help classify the images better.

# References

* Dataset we have used for this project

<https://www.dropbox.com/sh/sprchr4eizkgota/AAB946uvOy6FUMhkhtx9ksfza?dl=0>

* <http://www-users.cs.umn.edu/~cherian/ppt/MachineLearningTut.pdf>
* <http://www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html>
* [http://stats.stackexchange.com](http://stats.stackexchange.com/)
* <http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-CVPR-05.pdf\>
* class slides
* <http://scikit-learn.org/stable/supervised_learning.html#supervised-learning>
* <http://www.cs.unc.edu/~lazebnik/spring09/lec18_bag_of_features.pdf>
* <http://vhosts.eecs.umich.edu/vision//teaching/EECS442_2011/lectures/lecture19.pdf>
* http://ufldl.stanford.edu/eccv10-tutorial/