# Using SVM Learning With RBF Kernel for Image Classification

Group Name: YY
Group Member: Tianqi Wen, Mengyu Yang, Ding Zhao

## ABSTRACT

Feature extraction sees a wide range of applications in image classification. It enjoys advantages such as reducing computation power and helping eliminate over-fitting problems of certain classification algorithms while maintaining sufficient accuracy. In this project, we tried to experiment this approach on Support Vector Machine (SVM) and explored several variants: we experimented on combining Principal Component Analysis (PCA), PCA with K-Means clustering and Histogram of Oriented Gradients (HOG) with the baseline SVM-Radial Basis Function kernel method. We showed that HOG feature extraction could dramatically increase the performance while reducing the number of features to ⅔ of the original ones.

## KEYWORDS

SVM, RBF kernel, PCA, K-Means, HOG feature extraction

## 1. INTRODUCTION

During the previous phase of the project, our team experimented and developed Support Vector Machine with linear kernel, polynomial kernel & Radial Basis Function (RBF) kernel, Neural Network with 1 hidden layer as well as Multinomial Logistic Regression as the classifiers for the given dataset. In this final phase of the project, we evaluated and selected SVM with RBF kernel as the baseline method for further improvements. With the focus laid on data preprocessing and feature extraction, analysis tool such as Principal Component Analysis, K-Means clustering and Histogram of Oriented Gradients were employed. Statistical results as well as comparison observations were included in this report to illustrate why we finalized our classifier as HOG feature extraction with the baseline SVM-RBF kernel method.

## 2. BACKGROUND

Baseline algorithm extended in this phase is Support Vector Machine. The Support Vector Machine classifier is a binary classifier which looks for an optimal hyper-plane as a decision function. In section 3, experiments and comparisons on performance between 3 different SVM kernels are listed. The finalized kernel selection was RBF kernel, which is a function of the squared Euclidean distance between feature vectors and hence is also interpreted as a "similarity measure" [3]. We ran the SVM from starter code on the full training dataset in the first phase of the project. The best testing accuracy can achieve 37.2% with linear kernel and 42.7% with RBF kernel. Training took less than 3 minutes and stopped within 500 iterations. The training accuracy is around 70% - 90%.

Other methods experimented included Neural Network with 1 hidden layer and Multinomial Logistic Regression. The former was conducted based on revised code from 10601 HW3 and the latter was implemented according to project starter code. Due to comparatively low testing accuracies, these two methods were not considered as baseline methods.

## 3. SVM WITH DIFFERENT KERNELS

### 3.1 Performance of SVM with three different kernels

The experiment was first conducted on simple SVM with different kernels. Three most popular kernels – linear, polynomial and radial basis function (RBF) kernels were tested.

The classifier was trained on the full training dataset. Each image was represented by RGB values. Since the image was 32*32 pixels, each image had 32*32*3 = 3072 features.

After tuning with different parameters, the best performance for the three kernels during our experiments is summarized as follows:

| Kernel | Training Accuracy | Test Accuracy |
|---|---|---|
| Linear | 82.78% | 37.2% |
| RBF | 76.88% | 42.7% |
| Poly | 9.32% | 9% |

Fig1. Performance for SVM with different kernels

The polynomial kernel displayed accuracies lower than expectation on both training and testing dataset. Experiments on order of 2, 3, 4, 5 were conducted and and it was discovered that polynomial kernel under-fitted the data under all conditions. Further investigation indicated that the nature of the dataset should be considered accountable. One possible cause might be that the decision boundary is nowhere close to polynomials specified.

The linear kernel produced satisfactory test accuracy (37.2%) and high training accuracy (82.78%). The difference in between suggested high possibility of overfitting. This observation led to the conclusion that the linear kernel model displayed high variance on predicting and hence was unstable concerning performance.

The RBF kernel gave the best performance among the experiment results. After continuous experiments on parameter tuning, the best accuracy can be achieved at 42.7%. In-depth examination suggested that the error were mostly resulted from similarities between local objects within an image (such as bird and airplane) and the obscure boundary between object and background.

### 3.2 Tuning parameters for SVM with RBF kernels

The parameters we engineered were regularization term and RBF kernel width. The table below shows the summary of the parameter values and test accuracy.

| Regularization term (lambda) | RBF kernel width (gamma) | Test Accuracy |
|---|---|---|
| 0.5 | 5 | 23.1% |
| 0.5 | 1 | 36.8% |
| 0.5 | 0.1 | 38.0% |
| 0.5 | 0.01 | 41.9% |
| 0.5 | 0.001 | 36.5% |
| 0.5 | 0.007 | 40.7% |
| 0.07 | 0.007 | 42.7% |
| 0.007 | 0.007 | 40.6% |

Fig2. Parameter values vs. test accuracy for RBF kernel SVM

At first we fixed regularization term to be 0.5 and tuned kernel width. As can be seen from the chart, the test accuracy increased with the kernel width decreasing from 5 to 0.01. From 0.01 to 0.001 the test accuracy began to drop, so we chose a value 0.007

within this range and tuned the regularization term. The test accuracy increased first and then decreased with the regularization term going down from 0.5 to 0.007. At last we finalized the regularization term to be 0.07 which gave the best test accuracy of 42.7%.

After locating the best parameter and baseline model, we focused on preprocessing the data and feature extraction in the following experiments.

## 4. PCA & K-Means

First, we preprocessed the dataset using Principal Components Analysis (PCA) which would reduce each instance to a linear combination of principal components. Assumptions were made that when training/classifying on the preprocessed dataset, the model would concentrate on the more distinguishing features. However, only slight improvement (testing accuracy 43.15%) was observed after performing PCA (extracting the first 500 principal components, experiment producing the best testing accuracy) on the dataset.

One possible explanation of the poor improvement might be the low degree of correlation between the raw pixel data from dataset images. Examination on the component coefficients obtained after running PCA verified this conjecture, since the coefficient values were generally smaller than 0.03. Another possible limitation came from the low resolution of the dataset images. As could be observed directly, the CIFAR images displayed obscure object-background boundary. Hence, environmental noises such random illumination or shadowing could severely affect object identification.

Inspired from classmates' experience in milestone 1, we also experimented combining PCA with K-Means clustering. The strategy proposed was first employing PCA to reduce the raw data dimensions on the full dataset, then performing clustering on the extracted features of each image, finally reconstructing the images based on the cluster centroids. During experiments, it was observed that no significant enhancement in testing accuracy was produced. After self-examination, we figured that using PCA to reduce the clustering dataset might not be proper given the inapplicability of PCA mentioned earlier. Whether other possible solutions such as random patching might help remain for further investigation.

## 5. HOG FEATURE EXTRACTION

### 5.1 Replace Original Feature with HOG Feature

Feature extraction based on PCA did not outperform the baseline results in the way as expected. After intense online research, new algorithm was studied and verified. Histogram of Oriented Gradients (HOG) is widely used feature descriptors in object detection field. As the name suggests, this algorithm focuses on gradient orientation of local objects within an image. Its key advantage, high in-variance to geometric and photometric transformations outperforms PCA in this regard [4].

From the baseline performance, it could be observed that the original image pixel features did not show key distinguishability among the classes. With its ability to reflex the object shape, HOG features were proved to be desirable. Particularly in CIFAR-10 dataset, given the vague nature, shape could work better than color histogram, texture and other possible features. After feeding this preprocessed features to SVM, we then

ran SVM-RBF kernel with lambda = 0.05, gamma = 0.07.
The experiment results are as follows:

| Features | Accuracy |
|---|---|
| Original Feature (rgb value) | 0.42648 |
| Append HOG Feature (patch size 8x8) to Original (rgb) | 0.50952 |
| Append HOG Feature (patch size 8x8) to Original (gray) | 0.54248 |
| HOG feature (patch size 8x8) | 0.54248 |
| HOG feature (patch size 8x8 + 4x4) | 0.60914 |

Fig3. Accuracy after adding HOG features

As shown by the figure, the improvement after adding HOG features was remarkable. By generally increasing the proportion of HOG features comparing to the original ones, the accuracy smoothly increased from ~42% to ~61%.

### 5.2 Method

Since HOG extraction counts occurrences of gradient orientation in localized portions of an image, similarly to edge orientation histograms, we need to deal with the original size images instead of vectors.

Reshape of the instance to 32x32x3 was performed followed by the gray-scale transformation. Then extract HOG features with patch size 8x8. Reshape the HOG feature back to a vector and use it as input feature of SVM classifier.

### 5.3 Tuning parameters

Fig4 shows the trend of test accuracy when we fixed regularization term lambda as 0.7 and tune RBF kernel width gamma. Max accuracy found when gamma = 0.056.

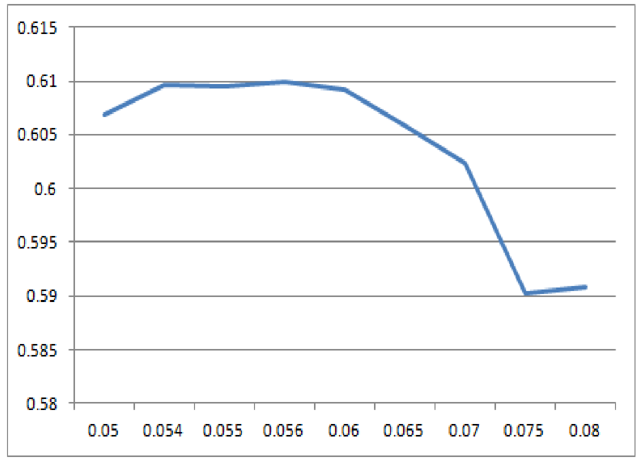


Fig4. Test accuracy vs. different gamma value. (lambda = 0.7).

Fig5 shows the trend of test accuracy when we fixed RBF kernel width gamma as 0.056 and tuned regularization term lambda. Max accuracy found when lambda = 0.005

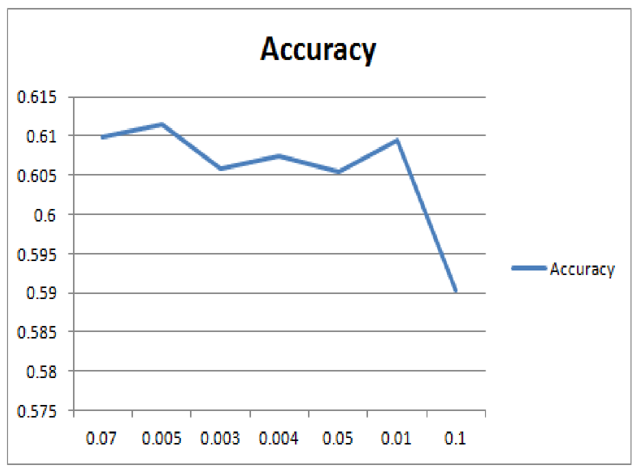


Fig5. Test accuracy vs. different lambda value. (gamma = 0.056).

## 6. CONCLUSIONS

Classification results from the previous phase of the project suggested error source as training over-fitting and poor object detection. Motivated by the advantages of feature extraction techniques in these two fields, we experimented combining SVM-RBF kernel with several feature extraction methods. Experimental results of PCA showed slight improvement in classification accuracy, which could be resulted from low degree of data correlation and vulnerability to environmental noises. Convolutional method PCA+K-Means was also conducted but was limited not only by the inapplicability of PCA. Last but not least, we tried the HOG feature extraction method, which extracted the shape feature and displayed high in-variance to geometric/photometric interference. A drastic increase in testing accuracy (from ~42% to ~61%) could be observed, and, together with the baseline SVM-RBF kernel classifier, it was considered as our finalized method.

## 7. REFERENCES

[1] http://www.cs.toronto.edu/_kriz/cifar.html

[2] http://www.kaggle.com

[3] Vert, Jean-Philippe, Koji Tsuda, and Bernhard Schölkopf (2004). A primer on kernel methods. Kernel Methods in Computational Biology.

[4] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection.