ImageNet Classification with Multiple Classifiers

Zheyun Feng
fengzhey@msu.edu

Jianpeng Xu
xujianpe@msu.edu

Abstract

In this project we proposed an ensemble classifier to classify over 20 thousand images sampled from ImageNet, which originally has over 10 million images. One of the challenge of this classification problem is that the images cannot be precisely represented by one type of features, such as SIFT and GIST. Hence, in this project, we use different kinds of features. Another challenge is that different classification models perform differently on different feature set. Here, we use different classifiers (Kernel Regression and SVM) on different features and ensemble the classification results in a weighting manner. The accuracy of the ensemble classifier outperforms almost all of the baselines.

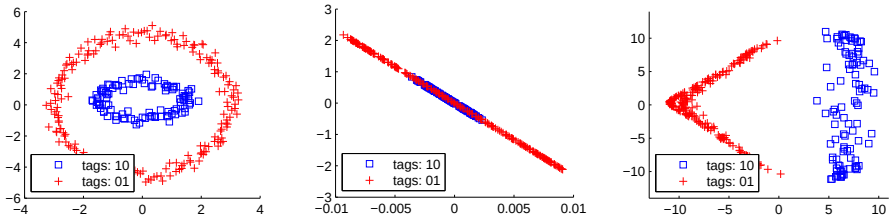
1 Introduction

With the development and popularity of image capturing devices and the emerge of large-scale storage, large-scale image classification has become more and more attractive. A lot of benchmark datasets have been formulated to fulfill the requirements of evaluating different image classification, clustering and retrieval algorithms. Some examplar popular benchmarks are Caltech 101 [8], PASCAL VOC [7], LabelMe [13], ImageNet [6], etc. Among those benchmarks, ImageNet dataset attract pretty much attentions because of its large number of images (over 1.2 million), and its hierarchical class structure which could generate up to 1000 classes. For the purpose of this project, a subset of the ImageNet dataset has been chosen, which is totally consisted of $25,000$ images, including 10 easy classes and another 10 difficult categories.

A number of descriptors have been proposed and proved to be effective in capturing the visual contents of images. Popular examples include LBP [14], SIFT [10], and GIST [12]. LBP descriptors convey the local information of each pixels in an image. SIFT descriptors describe the informative information like edges, corners and circles in an image and use a histogram statistics as a further extracted feature. This descriptor is scale invariant and is robust to rotations of images. GIST is a descriptor proposed specifically to represent the scene information using low dimension. Since these descriptors have their own perspectives of describing the image characteristics, using single one of them might not be able to fully describe an image. This inspires us to combine different descriptors together. However, it is not trivial on how to combine the descriptors and how to choose the classifier models. There are several possibilities. One is to concatenate different descriptors together. This approach is simple and sometimes with a good representation for images, but we do not use this scheme because of the different scales and significance of different type of descriptors. The descriptors with a large scale will dominate the feature space, and on the other hand some descriptors may be more informative than others and deserve a large significance weight. Another possibility is to train a classification model using each of the descriptors and then combine the output of these models. The challenge of this strategy is how to choose classification models for descriptors and how to combine the prediction results.

Many existing classification algorithms train a linear model for classifiers, which usually fail in capturing the intricate dependence among images that lie in a nonlinear manifold. Figure 1 illustrates an example comprised of two groups of data points, with each belonging to only one class. Figure 1(b)

shows the distributions of data points adjusted by a learned classifier, which fails in separating the objects with different class assignments.



(a) Original data (b) Linear classifier (c) Nonlinear classifier

Figure 1: An illustrative example for nonlinear classifiers. (a), (b), and (c) show the original data distribution, the distribution adjusted by a learned **linear** classifier, and the distribution adjusted by a learned **kernel** classifier, respectively.

A number of nonlinear classification algorithms have been proposed to overcome the limitations of linear classifiers. The key idea is to map data points from the original vector space to a high dimensional (or even infinite dimensional) space through a nonlinear mapping. Figure 1(c) shows that the two groups of data points, which is difficult to be separated by a linear classifier, can be well separated by using a learned kernel classifier. The mapping can be derived implicitly through the introduction of kernel function. Due to the advantages of nonlinear classifiers, in this project, we propose to use kernel tricks in our classifiers. And two classifiers are utilized for classification: Kernel Regression based classifier and SVM with RBF kernel.

The rest of the report is organized in the following way: in Section 2 some related work will be briefly introduced. Section 3 will presents the classifiers used in this project, including Kernel Regression based classifier and one-vs-one SVM. Section 4 shows the experimental results and analysis. Finally, Section 5 concludes this project and briefly gives out the future direction.

2 Related Work

One of the most popular algorithm to image classification is to use the bag-of-visual-words as features and apply a non-linear SVM [3]. Algorithms presented in [18] and [15] use a combination of different descriptors to train non-linear classifiers on corresponding descriptors and finally combine the output of the classifiers. Although these algorithms work well on PASCAL dataset, they are not efficient on large scale image set like ImageNet, as well as require large storage memory. To overcome these limitations of dealing with large scale data, [11] proposed an on-line version of SVM with a parallel implementation to speed up the training and reduce the memory. Instead of modifying the classifiers, [17] obtained good performance by using linear classifiers on sparse coding with a max-pooling of the descriptor-level statistics. And [4] proposed to use Fisher Vector as new feature representations. However, even though with Fisher Vector, it is possible to use a linear classifier to obtain a good prediction result, the Fisher Vector itself is with very large dimension and hence not suitable to be scaled up.

In our project, we use a subset of ImageNet, and hence are able to train non-linear classifiers effectively and efficiently. Furthermore, we choose an appropriate combination strategy to integrate different classifiers and improve the final classification performance.

3 Multiple Classifiers

In this project, we propose a classification algorithm which combines multiple classifiers, including our proposed kernel regression based classifier and the one-vs-one SVM classifier. Let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^\top$ be a set of training instances, where $\mathbf{x}_i \in \mathbb{R}^d$ is an image represented by a d-dimensional vector. Let m be the number of classes, and $Y = (\mathbf{y}_1, \cdots, \mathbf{y}_n)^\top$ be the class assignments of the training instances, where $\mathbf{y}_i \in \{0, 1\}^m$ with $y_{i,j} = 1$ if $\mathbf{x}_i$ is assigned to class j and 0,

otherwise. Since each image belongs to only one class, each $\mathbf{y}_i$ contains only one "1". Besides the sparse class assignment representation, we also use the dense representation $C = (c_1, \cdots, c_n)^\top$, where $c_i \in \{1, \cdots, m\}$. In this section, we first present the proposed *Kernel Regression* (KR) based classifier, followed by the SVM one-vs-one classifier.

3.1 Kernel Regression based Classifier

The proposed KR algorithm is a classification algorithm based on the regression techniques. Assume that the data points X are drawn independently from the distribution $p(y_{i,j}|\mathbf{x}_i, \mathbf{w}, \beta) = \mathcal{N}(y_{i,j}|y(\mathbf{x}_i, \mathbf{w}_j), \beta_j^{-1})$, and $y_{i,j} = y(\mathbf{x}_i, \mathbf{w}_j) + \epsilon_j$ where ϵ_j is a zero mean Gaussian random variable with precision (inverse variance) β_j. We extend the linear regression model by considering linear combination of fixed nonlinear functions of the input variables, of the form $y(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^{m} w_j \phi_j(\mathbf{x})$, where $\phi_j(\mathbf{x})$ are known as basis functions.

Let $\kappa(\mathbf{x}, \mathbf{x}')$ be the RBF kernel function that $\kappa(\mathbf{x}, \mathbf{x}') = \exp\{-\theta\|\mathbf{x} - \mathbf{x}'\|^2\}$, hence $\phi(\mathbf{x})$ could be equal to $[\kappa(\mathbf{x}, \mathbf{x}_1), \cdots, \kappa(\mathbf{x}, \mathbf{x}_n)]^\top$. We then obtain the following expression for the likelihood function, this is a function of the adjustable parameters $\mathbf{w}_j$ and β_j as follows

$$p([\mathbf{y}]_j|X, \mathbf{w}_j, \beta_j) = \prod_{i=1}^{n} \mathcal{N}(y_{i,j}|\mathbf{w}_j^\top \phi(\mathbf{x}_i), \beta_j^{-1}).$$

Note that in supervised learning problems such as regression or classification, we are not seeking to model the distribution of the input variables. Thus we get rid of the explicit expression of $\mathbf{x}$ and take the logarithm of the likelihood function

$$\ln p([\mathbf{y}]_j|X, \mathbf{w}_j, \beta_j) = \sum_{i=1}^{n} \ln \mathcal{N}(y_{i,j}|\mathbf{w}_j^\top \phi(\mathbf{x}_i), \beta_j^{-1}) = \frac{N}{2} \ln \beta_j - \frac{N}{2} \ln(2\pi) - \beta_j E_D(\mathbf{w}_j),$$

where the sum-of-squares error function is defined by

$$E_D(\mathbf{w}_j) = \frac{1}{2} \sum_{i=1}^{n} \{y_{i,j} - \mathbf{w}_j^\top \phi(\mathbf{x}_i)\}^2. \tag{1}$$

To determine $\mathbf{w}_j$, we can maximize the likelihood, equivalent to minimizing the sum-of-squares error function given by $E_D(\mathbf{w}_j)$, and get the close solution to (1) as follows

$$\mathbf{w}_j = (\Phi^\top \Phi)^{-1} \Phi^\top [\mathbf{y}]_j,$$

where $\Phi = [\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)]^\top$ represents the kernel space spanned by all the training images.

To make the above solution correspond to all the class assignments, we extend it to $W = (\Phi^\top \Phi)^{-1} \Phi^\top Y$, where $W = [\mathbf{w}_1, \cdots, \mathbf{w}_m)]^\top \in \mathbb{R}^{d \times m}$.

Furthermore, to avoid the overfitting risk, we add a regularization term r, and replace Φ with Φ_r, the best rank r approximation of Φ, and express W as

$$W = (\Phi_r^\top \Phi_r)^{-1} \Phi_r^\top Y = V_r \Sigma_r U_r^\top Y, \tag{2}$$

where $\Sigma_r = diag(\frac{1}{\sigma_1}, \cdots, \frac{1}{\sigma_r})$ and σ_i is the i-th top singular value of Φ, while U_r and V_r contain the top r left and right singular vectors of Φ, respectively.

Evidently, the rank r makes the tradeoff between bias and variance in estimating W: the larger the rank r, the lower the bias and higher the variance.

Finally, we obtain the confidence score $\mathbf{p} \in \mathbb{R}^m$ for a test image $\mathbf{x}_t$ as follows

$$\mathbf{p} = \Phi(\mathbf{x}_t) W = \Phi(\mathbf{x}_t) V_r \Sigma_r U_r^\top Y. \tag{3}$$

L_1 normalizing $\mathbf{p}$ by 1, we can obtain the probability of the test image $\mathbf{x}_t$ to be assigned to each class. Usually, the class corresponding to the maximum probability would be considered as the predicted class.

3.2 SVM one-vs-one Classifier

For a two-class classification problem, assume the two classes is linear separable, the SVM is trying to find a hyperplane that can give a maximum margin between two classes by optimizing the following objective function [2]:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2, \\ s.t. \quad & y_i(\mathbf{x}_i) \cdot \mathbf{x} + b - 1 \geq 0. \end{aligned}$$

However, in the real classification problem, it is not feasible to assume the two classes can always be linearly separable. Hence, some error can be tolerated while the objective is still maximizing the margin between two classes. The slack variables are introduced to represent the error that is can be tolerated. We call it the SVM with soft margin [2] and represent as follows

$$\begin{aligned} \max_{\mathbf{w}} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C(\textstyle\sum_i \xi_i)^k, \\ s.t. \quad & y_i(\mathbf{x}_i^T \cdot \mathbf{w}) + b - 1 \geq \xi_i, \end{aligned}$$

where ξ is the slack variables for each data points, and $\sum_i \xi_i$ is an upper bound of training errors. This optimization problem can be solved by adopting the Karush-Kuhn-Tucker (KKT)[1] conditions.

In practice, such as in image classification problem in this project, we are dealing with multi-classes rather than only two classes. Various methods are proposed to build multi-class classifiers by combine multiple two-class SVMs. Two of the most popular strategies are one-vs-rest SVM and one-vs-one SVM.

One-vs-rest SVM only needs to build $K-1$ classifiers for a K-class classification problem. However, this approach suffers from the problem that (1) the data becomes imbalance when taking data from a specific class as positive and data from all the other classes as negative; (2) the probabilities given by different SVMs might not be comparable because they are predicted by models trained for different objectives with different parameters.

One-vs-one SVM is to train totally $K(K-1)/2$ different two-class SVMs on all possible pairs of classes. The final class labels are determined by a voting mechanism. Even though this strategy needs to train much more models than the one-vs-rest strategy, it could avoid the problem of data imbalance. Hence, in our project, we use the one-vs-one SVM for the multi-class classification, although one-vs-one model still performs empirically well.

3.3 Classifier Combination Strategies

We tried three classifier combination strategies in this project, including majority voting, weighted majority voting and directly addition. For majority voting, we estimate the predicted class for each image using all classifiers, and the class predicted by the most classifiers is considered as the final predicted class for that image. Since we evaluate the top 5 predicted classes, we also tried the weighted majority voting scheme, where for each classifier, we assign each predicted class a weight based on its rank. For example, the predicted class with the top confidence score is weighted by 5, the one with the second top confidence score is weighted by 4 and the list goes on. Thus the prediction results from different classifiers could be added together, and the class with top score is regarded as the final predicted class. Finally, we also consider the scheme where the confidence scores of all classifier are directly added together. We obtain a confidence score matrix for each classifier P_k, which could be either a probability (in SVM) or regression results (in KR), then normalize the rows of these matrices using L_1 norm and obtain $\bar{P}_k$, and obtain the final confidence matrix P by adding them together with a weighted contribution as follows

$$P = \sum_{k=1}^{K} \alpha_k \bar{P}_k$$

where K is the number of applied classifiers, and α_k is the weight for the k-th classifier. If different features are applied to the same classifier, *e.g.*, both GIST and SIFT features are used to train SVM classifiers, we simply consider the two resultant models as different classifiers.

4 Experiments

4.1 Dataset and Experimental Setup

The original Imagenet data set consists of 10 million images downloaded from the web using keywords corresponding to more than $10,000$ classes. The dataset used in this project [1] is a subset of Imagenet 2012, which is consisted of $21,037$ training images, $1,057$ validation images and $4,194$ test images. Both SIFT descriptors [10] and GIST features [12], as well as the pixels of each image are provided. Each image has a single label out of 20 classes. The categories included are: geyser, odometer, canoe, yellow flower, website, gondola, rapeseed, flamingo, electric locomotive, daisy, ladle, hatchet, spatula, muzzle, hook, cleaver, letter opener, plunger, chimes, and power drill.

To achieve a good classification performance, we represent the images in the form of GIST features [12] along with combination of densely sampled SIFT descriptors [10]. The SIFT descriptors of all the images are quantized to $1,000$ visual words using k-means clustering algorithm, and a bag-of-words histogram is then generated for each image. GIST [12] features are obtained from the project description.

Even though this project deals with a classification problem, we still use five labels with the highest scores to calculate the classification accuracy. As long as one of these five labels matches the ground truth label, we would consider the classification as correct and assign the particular image with an error of 0. Only when none of these five labels matches the ground truth, we consider the classification fail and assign an error of 1.0. Finally, the average precision over all the test images are reported to evaluate the classification performance.

4.2 Final Experimental Results

Table 1 shows our final classification accuracy obtained by our proposed classification algorithm on training set, validation set and test set.

Training Accuracy	Validation Accuracy	Test Accuracy
99.95%	91.01%	90.10%

Table 1: Final accuracy obtained by our proposed algorithm.

Table 2 shows the confusion matrix of the classification results, form where we observe that some classes have high true positive rate, *e.g.*, Class 1, 5, 8 and 9, while some classes have low true positive rate, *e.g.*, Class 12, 13, 14 and 16, which correspond to hatchet, spatula, muzzle, and cleaver. As known, these classes have similar shapes even colors, and thus are difficult to distinguish.

Actually, our proposed classification algorithm combines three classifiers, including KR+SIFT classifier, KR+GIST classifier and one-vs-one SVM+GIST [2] classifier. Table 3 shows the total classification accuracy over the validation set when these three classifiers are used either individually or together, and Table 4 presents the classification accuracy for each class when using these three classifiers individually or together. From these two tables, we can easily conclude that the combination of classifiers could achieve better accuracy than any individual classifier. This may be because the high level class assignment of an image is determined based on many viewpoints, *i.e.*, color (yellow flower), shape (spatula), texture (website), *etc*., while a specific feature can only capture the content from one viewpoint. Even with the same feature, different classifiers would lead to different decisions for each class.

To combine the results of different classifiers, we simply adopt a parameter to the probability matrix of each classifier, and empirically select $\alpha = [0.97, 1, 1.64]$, corresponding to SIFT+KR, GIST+KR and GIST+SVM, respectively.

[1] Available at `http://www.cse.msu.edu/~cse802/datasets/imagenet/`.

[2] We directly use the LIBSVM library `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

		Predicted Class																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Actual Class	1	**100**	8	36	14	16	14	48	44	19	3	34	17	22	19	28	19	16	22	9	13
	2	12	**99**	36	18	31	13	8	23	15	15	50	10	40	22	28	9	10	10	24	27
	3	41	20	**93**	4	11	59	26	54	26	4	46	17	15	11	26	15	13	7	2	9
	4	10	12	12	**98**	4	8	20	61	6	73	22	10	37	41	24	0	4	29	12	14
	5	23	46	24	8	**100**	16	16	18	27	5	15	16	15	16	39	22	19	15	23	36
	6	28	9	62	6	15	**94**	6	26	43	9	40	4	30	11	32	9	2	21	28	28
	7	47	23	43	15	7	20	**97**	32	17	28	20	23	18	22	30	10	20	8	10	10
	8	16	28	46	54	4	22	20	**100**	8	36	38	24	18	26	12	6	0	20	18	4
	9	14	22	43	10	33	59	20	29	**100**	6	16	8	18	29	8	12	10	14	29	18
	10	14	21	9	55	3	15	39	36	2	**94**	44	9	32	27	29	12	8	21	9	21
	11	12	11	11	7	5	12	7	15	3	11	**93**	27	62	12	64	29	55	33	14	16
	12	15	6	12	0	9	6	6	9	0	0	59	**74**	59	12	79	35	65	21	15	21
	13	4	17	11	4	9	17	6	23	4	11	74	19	**79**	23	60	36	38	19	23	21
	14	3	16	19	14	3	14	19	41	3	22	59	19	43	**78**	57	16	8	24	16	27
	15	12	10	7	9	4	6	7	13	3	4	61	37	43	28	**90**	25	45	25	27	40
	16	8	5	11	3	3	8	0	13	0	5	68	34	82	11	71	**61**	47	26	18	26
	17	4	0	11	11	6	2	11	6	2	9	70	49	51	9	74	45	**85**	19	26	11
	18	8	5	8	11	3	19	11	22	3	8	62	19	62	24	51	24	19	**86**	27	27
	19	0	16	11	18	16	13	11	18	16	8	53	24	39	21	50	24	32	16	**82**	34
	20	7	18	5	9	20	4	4	13	0	11	41	27	36	52	61	21	23	32	32	**86**

Table 2: Confusion matrix (%) of the classification results, with diagonal line in bold.

SIFT+KR	GIST+KR	GIST+SVM	Combination Accuracy
82.78 %	86.00%	89.02%	91.01%

Table 3: Classification accuracy of three used classifiers before and after combination on the validation set.

Classifier	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SIFT+KR	95	**99**	83	**98**	**100**	91	93	**98**	96	**97**	75	62	60	57	73	47	**85**	54	55	**80**
GIST+KR	**100**	97	87	96	**100**	89	**98**	92	**100**	92	79	65	64	**84**	76	**58**	72	76	82	79
GIST+SVM	97	95	**91**	**98**	**100**	**94**	93	94	98	91	**93**	**71**	**83**	81	**91**	**58**	79	**78**	**84**	**80**
Combination	100	99	$\overline{93}$	98	100	94	$\underline{97}$	$\overline{100}$	100	$\underline{94}$	93	$\overline{74}$	$\underline{79}$	$\underline{78}$	$\underline{90}$	$\overline{61}$	85	$\overline{86}$	$\underline{82}$	$\overline{86}$

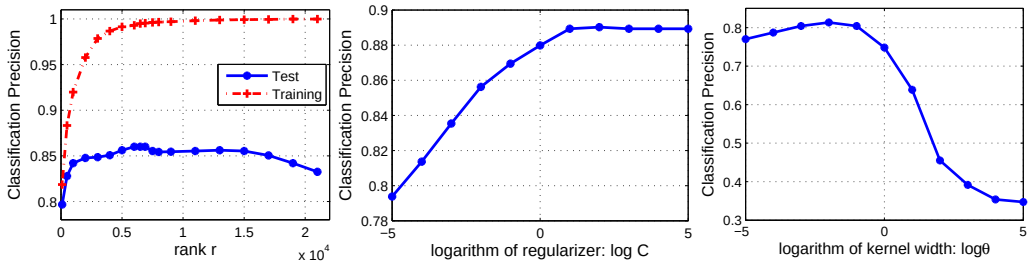
Table 4: Accuracy (diagonal of confusion matrix) (%) for each class using three classifiers as well as their combination on the validation set. The optimal values for each class among the three classifiers are in bold font. If the accuracy of combination excesses the optimal one, marked as $\overline{\cdot}$ and $\underline{\cdot}$, if no better than the optimal one.

4.3 Analysis of Parameters

We involve a kernel width θ and a regularized rank r in the Kernel Regression algorithm, and also a regularizer C and the kernel width g in SVM method. All of them are determined using 5-fold cross-validation in the project. In the following section, we present the influence of their choices. Note that when we analyze a certain parameter, we fix the other parameters to their optimal values.

From Figure 2(a), we observe that while the classification accuracy of the validation set initially improves significantly with increasing rank r, it becomes saturated after certain rank. On the other hand, the accuracy of training set increases monotonically with respect to the rank r, and becomes almost 1 for sufficiently large r, while the accuracy of validation set decreases apparently for too large r, clearly indicating the overfitting of training data.

Figure 2 (b) and (c) show how the classification precision changes with the regularizer C when fixing $g = 0.25$ and g when fixing $C = 4$. The optimal empirical parameters are $C = 4$ and $g = 0.25$ which is obtained by a grid search.



(a) rank r in KR (b) regularizer C in SVM (c) kernel width θ in SVM

Figure 2: Analysis of parameters, including rank r in KR (a), regularizer C and kernel width g in SVM. Dash lines represent the training accuracy and solid lines indicate the validation accuracy. For (b) and (c), the horizontal axis is scaled by logarithm, *i.e.*, $\log C$ and $\log \theta$, respectively.

4.4 Advantages of Nonlinearity

Linear Regression	Linear SVM [3]	Kernel Regression	RBF Kernel SVM
71.81%	79.56%	86.00%	89.02%

Table 5: Comparison of linear and kernel classifiers in terms of accuracy using GIST features on validation set.

Compare with the kernel classifiers used in our project, *i.e.*, the Kernel regression one and the RBF kernel SVM one, and their linear counterparts, it is clear that the kernel methods significantly outperform their linear counterpart, implying the nonlinearity of data.

4.5 Comparison with Other Methods

We compare our proposed algorithm to some other well-know algorithms, including one-vs-rest SVM (*SVM-all*), probabilistic models like Maximum likelihood method (*ML*) and Naïve Bayes method (*NB*), and K nearest neighbor methods with distance metric learning, including Euclidean distance (*KNN*), Discriminant Component Analysis (*DCA*) [9], Large Margin Nearest Neighbor classifier (*LMNN*) [16], and Information Theoretic based Metric Learning (*ITML*) [5]. Each algorithm is evaluated with both SIFT and GIST features on the validation set. To deal with the retrieved neighbors, weighted majority voting scheme is used, and the number of neighbors k is determined using cross-validation for each KNN method.

SVM-all	SVM-1	ML	NB	KNN	DCA	LMNN	ITML	KR	Proposed
85.53	89.02	23.46	23.46	80.70	78.15	81.27	77.48	86.00	91.01

Table 6: Comparison with well-known baselines in terms of classification accuracy (%) using GIST features. *SVM-1* represents the one-vs-one classifier used in our project, and *Proposed* indicating the proposed algorithm combining three classifiers.

Table 6 shows the classification accuracy compared with listed baselines. It is clear that the classifiers used in our algorithm as well as our combined classifier significantly outperform all the baselines.

5 Conclusion

In this project, we proposed an ensemble classifier which combines a Kernel Regression based classifier and one-vs-one SVM. SIFT and GIST descriptors are used and bog-of-words model is utilized. The final prediction is generated by combining the output of several models in a weighted manner. Our experiment results show our method outperforms most of the baseline algorithms such as ML,

KNN and SVM. In the future work, we plan to explore the classifier combination strategies, as well as search for more informative features. Beside, image segmentation techniques are also planed to adopt in order to locate the objective content in the images, which would effectively reduce the noisy descriptors extracted from the background and thus improve the classification performance.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, June 1998.

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[4] G. Csurka and F. Perronnin. In P. Richard and J. Braz, editors, *VISIGRAPP (Selected Papers)*, pages 28–42. Springer.

[5] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[6] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 71–84, Berlin, Heidelberg, 2010. Springer-Verlag.

[7] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[8] 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. *Comput. Vis. Image Underst.*, 106(1):59–70, Apr. 2007.

[9] S. Hoi, W. Liu, M. Lyu, and W. Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, 2006.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, 2006.

[11] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. S. Huang. Large-scale image classification: Fast feature extraction and svm training. In *CVPR*, pages 1689–1696, 2011.

[12] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[13] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *MIT AI Lab Memo*, 2005.

[14] O. T. and P. M. . M. T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971 - 987.

[15] M. A. Tahir, J. Kittler, K. Mikolajczyk, F. Yan, and T. Gevers. Visual category recognition using spectral regression and kernel discriminant analysis. In *In ICCV Workshop on Subspace Methods*, 2009.

[16] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advance in Neural Information Processing (NIPS)*, 2006.

[17] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *in IEEE Conference on Computer Vision and Pattern Recognition(CVPR*, 2009.

[18] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73:2007, 2007.