

MIAR ICT participation at Robot Vision 2013

Ruihan Xu, Shuqiang Jiang, Xinhang Song, Shuang Wang, Yi Xie, Fang Wang,
and Xiong Lv

Institute of Computing Technology, Chinese Academy of Sciences
No.6 Kexueyuan South Road Zhongguancun, Haidian District Beijing, China
{ruihan.xu, shuqiang.jiang, xinhang.song, shuang.wang, yi.xie, fang.wang}@vipl.ict.ac.cn, lvxiongforyou@foxmail.com

Abstract. This paper describes the participation of our team - MIAR ICT in the ImageCLEF 2013 Robot Vision Challenge. The task of the Challenge asked participants to classify imaged indoor scenes and recognize the predefined objects appeared in the imaged scene. Our approach is based on the recently proposed Kernel Descriptors framework, which is an effective representation for images. For the provided visual and depth sequences, we make a simple fusion at feature level. Then we use Linear Support Vector Machine (L-SVM) classifiers for both scene classification and object recognition. At last, the temporal continuity of the given sequences is considered. Our team ranked the first among all the participants, showing the effectiveness of our proposed scheme.

Keywords: kernel descriptor, scene classification, object recognition, temporal continuity

1 Introduction

In the 5th Robot Vision Challenge of the ImageCLEF 2013, image sequences were captured by a perspective camera and a Kinect[1] mounted on a mobile robot within an office environment. Visual (RGB) images and depth images generated from 3D point clouds were available. Training sequences were labelled not only with semantic labels (corridor, kitchen, office, etc.) but also with the objects that were represented in them (fridge, chair, computer, etc.). The test sequence were acquired within the same building and floor, but there could be variations in the lighting conditions (very bright places or very dark ones) or the acquisition procedure (clockwise and counter clockwise). Given test sequences, participants were asked to classify different indoor scenes, and judge the existence of the given objects within each image.

This paper describes the participation of our team in the Robot Vision Challenge. For the image features extraction part, we used the state-of-the-art Kernel Descriptors[2] framework, which has proven to be useful for many problems with RGB-D (visual and depth) information[9]. We applied L-SVM[11] for classification, and the temporal continuity is utilized during the classification stage.

The rest part of this paper is organized as follows. In Section 2, we briefly give a overview of our classification system. In Section 3, we describe in detail

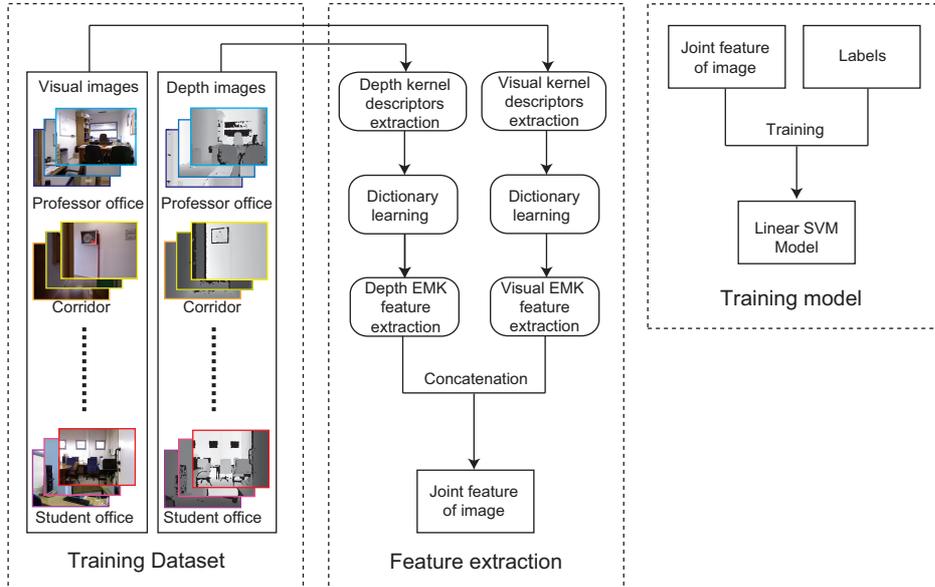


Fig. 1. Training stage of scene classification.

the image feature we used in our scheme. In Section 4 we describe how we apply classifier for both two tasks, and how we make use of the temporal continuity. In Section 5, we give some of our experiments and show our final result on test sequence. In Section 6, we draw some conclusions.

2 Overview

In this section, we describe the procedure of our scheme. Both scene classification and object recognition tasks can be solved using classification framework based on supervised learning. The training stage for scene classification is shown in Fig.1 and the test stage is shown in Fig.2. Framework for the recognition of each object is similar, except that training labels and predicted labels are replaced by the existence of each predefined object.

During the preprocessing stage of our scheme, the given 3D Point Cloud data are transformed to depth images, which afterwards will be treated as grayscale images. Then for both visual images and depth images, we extract Kernel Descriptors [2] as local descriptors, and use efficient match kernels (EMK) to transform and aggregate the descriptors to the features of images. We represent each frame by concatenating the two kinds of features extracted from each visual image and depth image. Then we choose L-SVM as our classifier for both scene classification and object recognition. In consideration of temporal continuity, we assign the averaged L-SVM scores of one frame's temporal neighbors to its final score. More details of our scheme will be described in the following sections.

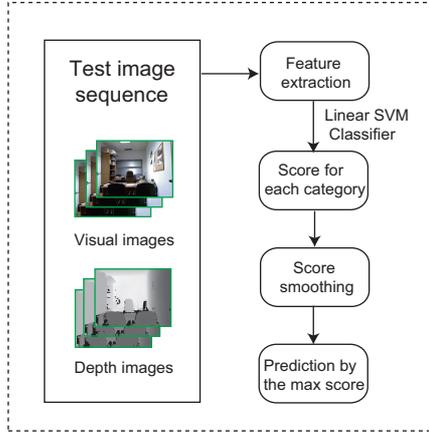


Fig. 2. Test stage of scene classification.

3 Image Features

In this section, we describe the features of image, which have been used in our work. The feature extraction procedure consists of two steps. The first step is to design match kernels using pixel attributes, and the second is to learn compact features.

3.1 Kernel descriptors

Kernel descriptors are able to generate rich patch-level features from different types of pixel attributes. For visual images, we use gradient, local binary pattern (LBP)[8] and color kernels. For depth images, we use depth gradient, depth LBP, spin/surface normal kernels.

The gradient match kernel is:

$$K_{grad}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \tilde{m}(p) \tilde{m}(q) k_o(\tilde{\theta}(p), \tilde{\theta}(q)) k_p(p, q), \quad (1)$$

where P and Q are the set of nearby points around the reference point \bar{p} and \bar{q} , respectively. $k_p(p, q) = \exp(-\gamma_p \|p - q\|^2)$ is a Gaussian position kernel with z denoting the 2D position of a pixel in an image patch (normalized to $[0, 1]$), and $k_o(\theta(p), \theta(q)) = \exp(-\gamma_o \|\theta(p) - \theta(q)\|^2)$ is a Gaussian kernel over orientations.

Kernel view of orientation histograms provides a way to turn pixel attributes into patch-level features, which can also be extended to LBP match kernel:

$$K_{LBP}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \tilde{s}(p) \tilde{s}(q) k_b(b(p), b(q)) k_p(p, q), \quad (2)$$

where $\tilde{s} = s(p) / \sqrt{\sum_{p \in P} s(p)^2 + \epsilon_s}$, $s(p)$ is the standard deviation of values in the 3×3 neighborhood around p , ϵ_s is a small constant, and $b(p)$ is a binary column vector which binarizes the pixel value differences in a local window around p .

Similar to gradient and LBP kernels, the color match kernel can be formulated as:

$$K_{col}(P, Q) = \sum_{p \in P} \sum_{q \in Q} k_c(c(p), c(q)) k_p(p, q), \quad (3)$$

where $c(p)$ is the pixel color at position z (intensity for gray images and RGB values for color images). $k_p(c(p), c(q)) = \exp(-\gamma_c \|c(p) - c(q)\|^2)$ measures how similar two pixel values are.

Since depth images are treated as grayscale images, depth gradient and depth LBP kernels are constructed in a similar way like the gradient and LBP kernels for visual images. Here we just describe another one of the depth kernels - spin/surface normal kernel[3].

In spin images[6], a reference point in a local 3D point cloud is represented as the pair (\bar{p}, \bar{n}) formed by its 3D coordinate \bar{p} and surface normal \bar{n} . The spin image attribute of a point $p \in P$ represented by the pair (\bar{p}, \bar{n}) is given by the triple $[\eta_p, \varsigma_p, \beta_p]$, where the elevation coordinate η_p is the signed perpendicular distance from the point p to the tangent plane defined by the pair (\bar{p}, \bar{n}) , the radial coordinate ς_p is the perpendicular distance from the point p to the line through the normal \bar{n} , and β_p is the angle between the normals n and \bar{n} . The point attributes $[\eta_p, \varsigma_p, \beta_p]$ can be aggregated into local shape features by the following kernel:

$$K_{spin}(P, Q) = \sum_{p \in P} \sum_{q \in Q} k_a(\bar{\beta}_p, \bar{\beta}_q) k_{spin}([\eta_p, \varsigma_p], [\eta_q, \varsigma_q]), \quad (4)$$

where $\bar{\beta}_p = [\sin(\beta_p), \cos(\beta_p)]$, P is the set of nearby points around the reference point \bar{p} . Gaussian kernels k_a and k_{spin} are used to measure the similarities of attributes β , η and ς , respectively.

3.2 Learning Compact Features

Evaluating kernels is computationally expensive when image patches are large. For both computational efficiency and representational convenience, the feature can be extracted as following:

1. uniformly and densely sample sufficient basis vectors from support region to guarantee accurate approximation to match kernels.
2. learn compact basis vectors using kernel principal component analysis.

EMK combines the advantage of both bag-of-words and set kernels. Here we briefly describe how the EMK transforms kernel descriptors to low dimensional space to achieve compact features (see [4] for details).

Take feature based on gradient match kernel for example, other kinds of feature can be extracted in the same way. Rewriting the Eq.1:

$$\begin{cases} k_o(\tilde{\theta}(p), \tilde{\theta}(q)) = \phi_o(\tilde{\theta}(p))^\top \phi_o(\tilde{\theta}(q)), \\ k_p(p, q) = \phi_p(p)^\top \phi_p(q) \end{cases}, \quad (5)$$

the feature over image patches will be:

$$F_{grad}(P) = \sum_{p \in P} m(p) \phi_o(\tilde{\theta}(p)) \otimes \phi_p(p). \quad (6)$$

where \otimes is the Kronecker product. A straightforward way to dimension reduction is to sample sufficient image patches from training images and perform KPCA for match kernels.

Sufficient Finite-dimensional Approximation Finite-dimensional features can be learned by projecting $F_{grad}(P)$ into a set of basis vectors. A key issue in this projection process is how to choose a set of basis vectors which makes the finite-dimensional kernel approximate well the original kernel. Given a set of basis vectors $\{\varphi_o(x_i)\}_{i=1}^{d_o}$ where x_i are sampled normalized gradient vectors, a infinite-dimensional vector can be approximated by a infinite-dimensional vector $\varphi_o(\theta(p))$ by its projection into the space spanned by the set of these d_o basis vectors. Such a procedure is equivalent to using a finite-dimensional kernel:

$$\tilde{k}_o(\tilde{\theta}(p), \tilde{\theta}(q)) = k_o(\tilde{\theta}(p), X) [K_o^{-1}]_{ij}^\top k_o(\tilde{\theta}(q), X), \quad (7)$$

which can be rewritten as:

$$\tilde{k}_o(\tilde{\theta}(p), \tilde{\theta}(q)) = [Gk_o(\tilde{\theta}(p), X)]^\top [Gk_o(\tilde{\theta}(q), X)]. \quad (8)$$

Here $k_o(\tilde{\theta}(p), X) = [k_o(\tilde{\theta}(p), x_1), \dots, k_o(\tilde{\theta}(p), x_{d_o})]^\top$ is a $d_o \times 1$ vector, K_o is a $d_o \times d_o$ matrix with $K_{oij} = k_o(x_i, x_j)$, and $K = G^\top G$. The resulting feature map $\varphi_o(\theta(p)) = Gk_o(\theta(p), X)$ is now only d_o -dimensional.

Compact Features The size of basis vectors can be further reduced by performing kernel principal component analysis over joint basis vectors:

$$\{\varphi_o(x_1) \otimes \varphi_p(y_1), \dots, \varphi_o(x_{d_o}) \otimes \varphi_p(y_{d_p})\}, \quad (9)$$

where $\varphi_p(y_s)$ are basis vectors for the position kernel and d_p is the number of basis vectors. The t -th kernel principal component can be written as:

$$PC^t = \sum_{i=1}^{d_o} \sum_{j=1}^{d_p} \alpha_{ij}^t \phi_o(x_i) \otimes \phi_p(y_j), \quad (10)$$

where α_{ij}^t is learned through kernel principal component analysis[10].

Under the framework of kernel principal component analysis, the gradient kernel descriptor for the patch P has the form:

$$F_{grad}(P) = \sum_{i=1}^{d_o} \sum_{j=1}^{d_p} \alpha_{ij}^t \left\{ \sum_{p \in P} \tilde{m}(p) k_o(\tilde{\theta}(p), x_i) k_p(p, y_j) \right\}. \quad (11)$$

With patch descriptors available, we apply bag-of-words model and spatial pyramid [7] to get the final representation of images. The details of parameter setting will be discussed in Sec.5.2.

4 Classification

4.1 Classifier

In our work, we applied the LibLinear[5] as our classifier, since SVM is widely used for classification task and performs effective especially when the scale of data is small. For scene classification, we train a multiclass one-vs-all L-SVM classifier. As there are 10 different concepts of indoor scene, 10 binary L-SVMs are trained for each concept. For object recognition task, we treat the existence of each object as a binary classification problem. Frames that contain the predefined object are taken as positive samples, and the rest are taken as negative samples.

For better comprehension, let us introduce some notation here. Let I_n be one image of the test sequence, $n \in \{1, 2, 3, \dots, N\}$, where N is the number of all images in the test sequence.

For scene classification, let S_n^c be the L-SVM output score for test image I_n on concept c , $c \in \{1, 2, 3, \dots, C\}$, where C is the number of concepts, and $C = 10$ in this task. Then the predicted label c_n^{pred} of a test image I_n is decided following the rule below:

$$c_n^{pred} = \underset{c}{argmax} S_n^c. \quad (12)$$

For recognition of object obj_k , $k \in \{1, 2, 3, \dots, K\}$, where K is the number of objects to be recognized, and $K = 8$ in this task. Let $S_{n,k}$ be the L-SVM output score of test image I_n for obj_k , and $c_{n,k}^{pred}$ indicates the predicted concept of I_n for obj_k , where $c_{n,k}^{pred} \in \{-1, 1\}$, -1 for concept absence and 1 for concept occurrence. Whether a certain kind of object exists in the test image can be judged as below:

$$c_{n,k}^{pred} = \begin{cases} 1 & S_{n,k} > 0 \\ 0 & S_{n,k} = 0 \\ -1 & S_{n,k} < 0 \end{cases}, \quad (13)$$

where the prediction 0 in Eq.(13) means that whether the object exists or not is ambiguous, and we deal with this situation with not classifying it. This happens only when $S_{n,k} = 0$, which means that for the test image I_n , object obj_k has the same confidence on both concept occurrence and absence.

4.2 Consideration of Temporal Continuity

Since all the images in training and test sequences are captured continuously, it is reasonable and feasible to make full use of the temporal continuity. In our work, we apply a smoothing method for the L-SVM score to improve the classification performance.

We empirically think that the concept of an image is quite likely the same with that of its temporal neighbors, and the L-SVM score should be less changed compared to its neighbors. Based on this assumption, we smooth the L-SVM score for both scene classification and object recognition task as below:

$$S_n^c = \frac{1}{2r+1} \sum_{k=n-r}^{n+r} S_k^c, \quad (14)$$

where r is the radius of smooth window. Eq.(14) indicates that the final L-SVM score for image I_n on a certain concept c is determined by all the scores of neighbors within the smoothing window. With all the L-SVM scores updated, we do classification and recognition on the basis of these new scores.

Choosing appropriate r is very important, for it has a relevance with the specific data and differs from scene classification and object recognition. The details of choosing r will be discussed in Sec. 5.3.

5 Experiments

5.1 Datasets and experimental setup

For Robot Vision Challenge this year, two training sequences are provided with 1947 and 3316 images respectively. An additional (labelled) validation sequence with 1869 images is also provided. The final test sequence involves 3315 unlabelled images. For all the sequences, RGB images and Point Cloud Data (PCD) are available. As mentioned in Sec.1, there can be variations in the lighting conditions or the acquisition procedure between test sequence and training sequences, and the validation sequence is similar to the test to some degree.

For depth features extraction, we transformed all the given PCDs into depth images, and crop the useless blank border. See the example in Fig.3.

For the scene classification task, we pick the same size of images for each class in the training sequences to avoid the imbalance between semantic classes. In our work, we pick out the concept with minimum training data, count the number of training images in the concept and set this number as the size of training data for all the other concepts.

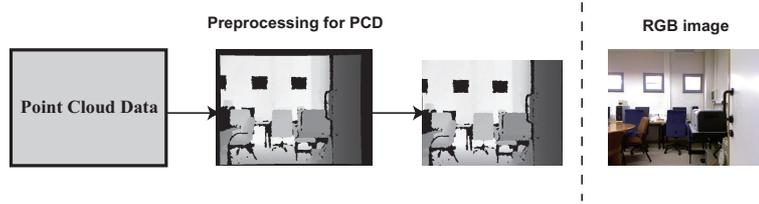


Fig. 3. (left) PCDs are transformed into depth images without blank border, (right) the RGB image is shown here for reference.

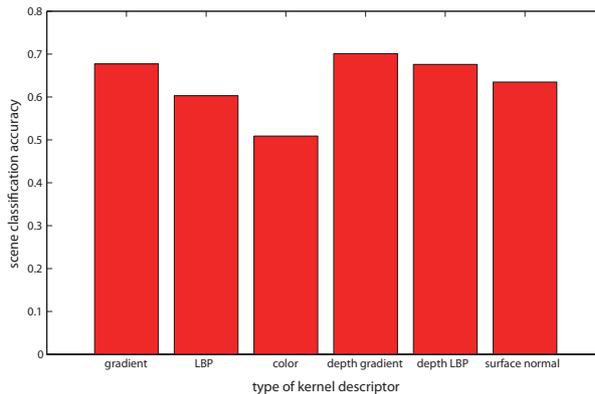


Fig. 4. Evaluation of 6 kernel descriptors on validation sequence.

5.2 Kernel descriptors selection

To choose appropriate kernel descriptors, we evaluate 6 kinds of kernel descriptors (gradient, LBP, color, depth gradient, depth LBP, spin/surface normal) through scene classification task on validation sequence. Fig.4 shows the evaluation of all the 6 kernel descriptors. Depth kernel descriptors perform a little better than visual kernel descriptors, due to the variations in the lighting conditions between the training sequences and validation sequence. For the final test, we select the three descriptors (gradient, depth gradient, depth LBP) which get the highest classification accuracy on validation sequence, since the test sequence is similar with validation sequence.

After 3 optimal kernel descriptors are chosen, we apply spatial pyramid (1, 2×2, 3×3) and perform the EMK transform with 1000 words. Then we get the image feature with a total length of $1000 \times (1 + 2^2 + 3^2) \times 3 = 42000$.

5.3 Smoothing window radius

As explained in Sec.4.2, the radius of smoothing window is important for the final performance. We perform experiments on different radius r for validation se-

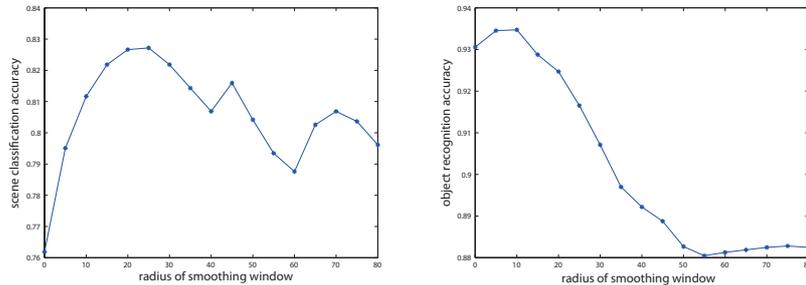


Fig. 5. (left)Variation of scene classification accuracy with r on validation sequence. (right)Variation of object recognition accuracy with r .

quences, and choose the r which corresponds to the highest accuracy. Fig.5 (left) shows how the scene classification accuracy varies with the radius of smoothing window r . We set the step width as 5, and find out that the accuracy reaches the climax when r is around 25.

Since the radius is related to the length of continuous scene images with the same concept, and there is a proportion between the quantity of validation images and test images, the estimated radius r should be multiplied the proportion to fit the test sequence. According to Eq.15, we get the estimated radius $r_{test}^{scene} = 44$ for test sequence.

$$r_{test}^{scene} = r_{validation}^{scene} \times \frac{N_{test}}{N_{validation}}. \quad (15)$$

For object recognition, Fig.5 (right) shows how the object recognition accuracy varies with the radius of smoothing window r , and it reaches the climax when r is around 10. Then we use the same method to get the estimated radius $r_{test}^{object} = 18$ for test sequence.

5.4 Results on validation sequence

We applied our method on the validation sequence, and computed the classification accuracy for each scene concept and object category. See Table 1 and Table 2 for detailed results. We notice that our method has good performance on each concept except 'StudentOffice' and 'TechnicalRoom'. This is due to the large variation of luminance between training and validation sequences on these two concepts images.

Scene Concept	Corridor	Hall	ProfessorOffice	StudentOffice	TechnicalRoom	Toilet
Accuracy	0.986	*	0.850	0.472	0.453	0.979
Scene Concept	Secretary	VisioConference	Warehouse	ElevatorArea	Total	
Accuracy	0.934	*	*	1.000	0.827	

Table 1. Scene classification results on validation sequence. Here * means that there is no images with concept Hall, VisioConference or Warehouse in the validation sequence.

Object Category	Extinguisher	Computer	Chair	Printer	Urinal	Fridge	Screen	Trash	Total
Accuracy	0.949	0.884	0.907	0.899	0.935	0.998	0.919	0.960	0.935

Table 2. Object recognition results on validation sequence.

5.5 Results on Robot Vision task

In the 5th edition of the Robot Vision Challenge, Our team ranked the first out of six participants, results are listed in Table 3.

#	Group	Score Class	Score Objects	SCORE TOTAL
1	MIAR ICT	3168.5	2865.000	6033.500
2	NUDT	3002.0	2720.500	5722.500
3	SIMD*	1988.0	3016.750	5004.750
4	REGIM	2223.5	2414.750	4638.250
5	MICA	2063.0	2416.875	4479.875
6	GRAM	-487.0	0.000	-487.000

Table 3. Robot Vision final results.

6 Conclusion

In this paper we present our scheme on the 5th Robot Vision Challenge. Our approach leverages the state-of-the-art methods in the fields of RGB-D image classification. Among all the participants for the Challenge, our team ranked the first, showing the effectiveness of our approach. Since the predefined objects have high dependence on indoor scenes, we achieve high accuracy on object recognition by using the representation of whole image. This method is useful for the specific task of this Challenge, and we plan to investigate more effective methods to better tackle general object recognition problem in future work.

Acknowledgements

This work was supported in part by National Basic Research Program of China (973 Program): 2012CB316400, in part by National Natural Science Founda-

tion of China: 61070108, and 61035001, in part by the Key Technologies R&D Program of China under Grant no. 2012BAH18B02.

References

1. *Microsoft Kinect*. <http://www.xbox.com/en-us/kinect>.
2. Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. *Advances in Neural Information Processing Systems*, 7, 2010.
3. Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 821–826. IEEE, 2011.
4. Liefeng Bo and Cristian Sminchisescu. Efficient match kernel between sets of features for visual recognition. *Advances in neural information processing systems*, 2(3), 2009.
5. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.
6. Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.
7. Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.
8. Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.
9. Xiaofeng Ren, Liefeng Bo, and Dieter Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012.
10. Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
11. Vladimir Vapnik. *The nature of statistical learning theory*. springer, 1999.