JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering
Vol.00, No.0, Jan 0000, pp. 000-000

Relation based Bayesian Network for NBNN

Mingyang Sun
School of Computing, KAIST, Korea **mingyang.sun@kaist.ac.kr**

YoonSeok Lee
School of Computing, KAIST, Korea **ys.lee@kaist.ac.kr**

Sung-eui Yoon*
School of Computing, KAIST, Korea **sungeui@gmail.com**

Abstract

Under the conditional independence assumption among local features, the Naive Bayes Nearest Neighbor (NBNN) classifier has been recently proposed and performs classification without any training nor quantization phases. While the original NBNN shows high classification accuracy without adopting an explicit training phase, the conditional independence among local features is against the compositionality of objects indicating that different, but related parts of an object appear together. As a result, the assumption of the conditional independence weakens the accuracy of classification techniques based on NBNN.

In this work, we look into this issue, and propose a novel, Bayesian network for NBNN based classification to consider the conditional dependence among features. To achieve our goal, we extract a high-level feature and its corresponding, multiple low-level features for each image patch. We then represent them based on a simple, two-level layered Bayesian network, and design its classification function considering our Bayesian network. To achieve low memory requirement and fast query-time performance, we further optimize our representation and classification function, named relation-based Bayesian network, by considering and representing relationship between a high-level feature and its low-level features into a compact relation vector, whose dimensionality is same to the number of low-level features, e.g., four elements in our tests. We have demonstrated benefits of our method over the original NBNN and its recent improvement, local NBNN, in two different benchmarks. Our method shows improved accuracy, up to 27% point against the tested methods. This high accuracy is mainly thanks to considering the conditional dependences between high-level and its corresponding low-level features.

Category: Smart and Intelligent Computing

Keywords: Naive Bayes nearest neighbor, classifications, conditional dependency, Bayesian network

I. INTRODUCTION

Image classification is one of main research challenges in the field of computer vision. Its goal is to identify a category of a query image based on a classifier. Many classification techniques have been proposed and can be roughly divided into two families. The first one is learning-based (parametric) methods such as SVM [1], which use a certain model and may require an intensive learning/training phase for optimizing classifier parameters of the model. Another one is non-parametric methods such as nearest neighbor based classifiers that work directly on image data and may not require a learning/training phase. These two different techniques have different strengths, but learning based approaches typically achieve higher accuracy over non-parametric approaches, mainly because of its learning phase. Nonetheless, parametric approaches can work poorly when the employed model does not fit well with data under the study [2].

Recently, Boiman et al. [3] proposed an efficient, yet accurate non-parametric method, the Naive Bayes Nearest Neighbor (NBNN) classifier, for image

Open Access yy.5626/JCSE.2011.5.2.xxx http://jcse.kiise.org

Received NA, Accepted NA, Revised NA

* Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

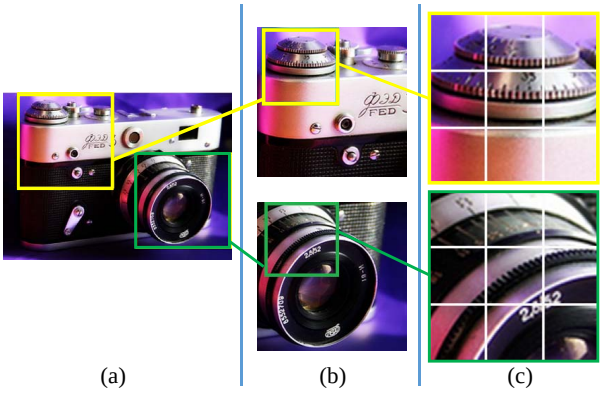

Fig. 1. The camera (a) is constructed by several parts (b), whose local image features (c) are correlated with each other. To capture such conditional dependence, we use a two-level layered Bayesian network for the NBNN approach. (c) shows 3 by 3 image regions of a part of the object.

classification. Its main idea is that given a set of local image descriptors extracted from a query image, NBNN identifies the class of the query image based on simple schemes such as using the image-to-class distance without quantizing (e.g., bag-of-visual-words) descriptors that can lose original information encoded in the input, unquantized descriptors. In spite of the simplicity and the absence of learning/training phases in NBNN, it has been reported to achieve surprisingly remarkable performance in image classification.

NBNN has drawn attentions from researchers and has generated many follow-up studies. At a high level, recent studies have identified that the original NBNN has high computation complexity during testing query images, works mainly in balanced datasets, and has an unrealistic assumption of the independence among features. Some of these issues have been addressed by adopting a parametric version of the NBNN with a training step [4], class-to-image distance [5], the NBNN kernel [6], and local NBNN [7]. Nonetheless, the conditional independence assumption has been understudied for the NBNN classification techniques according to the best of our knowledge.

For images, we typically extract various local features (e.g., dense SIFTs [8]) for image classification. For example, Fig. 1 shows conditional dependence among such local features. The camera object consists of different parts, each of which is represented by multiple features. This is so-called compositionality of objects [9]. Multiple features located locally can be grouped together to represent one part of the object, and several parts can be merged together to represent the object. As a result, the compositionality of objects can be represented by multiple layers. To consider such conditional dependence of local features, Naive Bayes can be transformed to a Bayesian network, where the Naive Bayes is the simplest form of Bayesian networks [10]. Unfortunately, extracting the optimal dependence among local descriptors for Bayesian networks is an NP problem [11], and thus has raised significant technical challenges.

To consider the conditional dependence without requiring intractable time complexity, we propose to extend the original NBNN to a simple Bayesian network, two-level layered Bayesian network. For considering the conditional dependence, we extract two different types of local features, a single high-level and multiple low-level features for an image patch. We identify conditional dependence between the high-level and low-level features, and represent them in a classification function in a similar spirit to the original NBNN. To achieve a low memory requirement and computational overhead, we optimize our representation by capturing relationship between the single high-level feature and multiple low-level features in a low-dimensional, relation vector, and reformulate our classification function with the relation vector. One can treat our relation vector as a self-correlation descriptor between the high-level feature and its low-level features.

We have implemented our ideas and demonstrated that our method, relation-based Bayesian network, achieves significantly improved accuracy over the original and local NBNNs. Furthermore, our method has similar computational and memory requirement to that of the original NBNN. These results are achieved, mainly because we consider the conditional dependence between high-level and low-level features based on our classification function with compact, yet effective relation vectors.

II. BACKGROUND AND RELATED WORK

In this section, we review prior techniques directly related to our method.

A. Naive Bayes Nearest Neighbor (NBNN) Classifiers

NBNN techniques [3, 7, 6] are simple and intuitive for providing accurate classification results. Given a query image, Q, the original NBNN assigns Q to one class, $\hat{C}$, based on the Maximum Likelihood (ML) estimation among a possible class set indexed by C:

$$\hat{C} = \underset{C}{\operatorname{argmax}}\, P(C|Q). \tag{1}$$

Assuming a uniform prior over all classes and applying Bayes' rule, Eq. 1 can be transformed to:

$$\hat{C} = \underset{C}{\operatorname{argmax}} \log(P(Q|C)). \tag{2}$$

Let $\{d_1, d_2, \ldots, d_n\}$ denotes all the descriptors extracted from the query image Q, which are assumed to be conditionally independent. Under the independence assumption, the ML estimation for

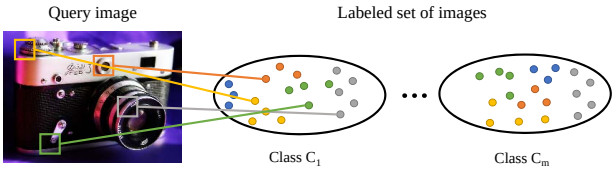

Fig. 2. Naive Bayes Nearest Neighbor (NBNN) classifier. For each class, it identifies the nearest neighbor of each descriptor extracted from the query image, assuming the conditional independence among features.

NBNN can be transformed as follows:

$$\hat{C} = \underset{C}{\operatorname{argmax}} \log \left[\prod_{i=1}^{n} P(d_i|C) \right] \tag{3}$$
$$= \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log P(d_i|C) \right].$$

NBNN takes the Parzen Kernel estimator to compute the posterior probability for classifying images. As a result, the classifier is approximated as the following:

$$\hat{C} = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2, \tag{4}$$

where $NN_C(d_i)$ is the nearest neighbor feature of d_i among features extracted from all the images assigned to the class C. Fig. 2 visualizes the NBNN method with its data for each class.

In order to optimize the NBNN classifier for large-scale image classification, different techniques have been proposed. McCann et al. proposed the local NBNN by searching the nearest neighbors of the query image descriptors among the whole dataset to achieve the result in a much shorter elapsed time [7]. In [4], isotropic kernel bandwidths are introduced to reduce the bias in case of the unbalanced datasets with a training step. Tuytelaars et al. [12] proposed a NBNN kernel that can be combined with other kernels, and Wang et al. [5] proposed a class-to-image distance for multi-labeled image classification. Vitaladevuni et al. [13] applied a NBNN method to image categorization and video event detection.

In this paper, we look into the independence assumption among features, which has not been studied for NBNN techniques. Especially, we take an eye on the problem caused by the conditional independence assumption for image classification. Even before used in computer vision, NBNN classification techniques have been used in the text classification under the assumption that all the words in one document are independent. Nonetheless, this assumption is incorrect in many real-world problems. To address this issue, excellent improvements have been proposed in the text classification in the last decade. Inspired by these prior techniques, we aim to design a more accurate, yet efficient Bayesian classifier for image classification. Furthermore, we design our technique such that it can be combined with most prior techniques such as local NBNN and other distance metrics, and can achieve better accuracy over prior NBNN techniques.

B. Dependences among Local Features

A set of local features is typically extracted to represent an image. Many techniques as image representations have been proposed. Scale-Invariant Feature Transform (SIFT), for instance, is an algorithm proposed by Lowe [14] to detect or describe local features as key points. While it has been extremely successful, it can omit some discriminative features of one object, because some discriminative features do not appear in key points. For example, in Fig. 3, the feature d_1 is a discriminative feature of the camera because it has distinctive patterns that can be found on the lens of cameras, but it may not be detected as a key point. To address this issue, dense SIFT, which does not have scale and location selection, is adopted to achieve higher discriminative power for extracted features [8]. These features are produced on a regular grid or locations using a constant patch size with the same scale.

Suppose for Fig. 3 that local features of $d_1, d_2, \ldots, d_9$ are extracted from a part, D, of the camera based on the dense SIFTs. Also, suppose that d_1, d_2, d_3, d_4 are nearest neighbors to each other such that they can be grouped together to represent a small image region, D', which is a part of D. In this example of the camera class, the combination of d_1 and d_2 has a higher probability to decide the image category than having each one of d_1 or d_2. Furthermore, having two features of d_1 and d_2 can result in a higher probability to detect the image category than having both features of d_1 and d_9. Note that the feature d_9 shows a reflected object on the lens and is not a part of the camera object.

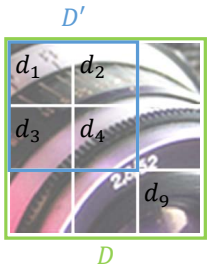

Fig. 3. The combination of d_1 and d_2 of a small region $D' \subset D$ can have a high probability to decide the image category.

As pointed out in the prior example, the dependences and correlations among near local features exist, and the combination of local features can help us to classify images better. It also shows that the assumption of conditional independence among all the local features can weaken the performance of the NBNN classifier. To address this issue, we build a layered structure of features with different scaled descriptors, and utilize the local dependences and correlations to gain better classification results.

C. Bayesian Network

Naive Bayes is the simplest form of Bayesian networks [15, 16, 17, 18], and has only two layers: one parent node and other nodes. Its simple structure is constructed on the assumption that given the parent node, the other nodes are independent, which is called *conditional independence*.

Bayesian networks are structured and graphical models that can represent relations between several random variables. In a Bayesian network, each variable is conditionally independent of all its non-descendants given the value of all its parents. This translates into the following equation:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i)), \quad (5)$$

where $X_1, \ldots, X_n$ are variables in the Bayesian network, and $parents(X_i)$ indicates parent nodes of X_i. In order to tackle the conditional independence assumption in NBNN, we use a simple Bayesian network considering conditional dependence among features to compute the probability.

Given the casual Markov assumption, we have the d-separation property [15, 16, 17, 18] related to conditional dependence and independence among features. For example, suppose that A and B are conditionally independent under parent node X. Similarly, suppose that $a_1, \ldots, a_4$ are conditionally independent under their parent node A, and $b_1, \ldots, b_4$ are too under their parent node B. We can then call that X and $\{a_1, \ldots, a_4 \cup b_1, \ldots, b_4\}$ are d-separated by A and B. The probability of having X in this context can then be computed as follows:

$$P(X, A, B, a_1, \ldots, b_4) = \quad (6)$$
$$P(X)P(A, B|X)P(a_1, \ldots, a_4|A)P(b_1, \ldots, b_4|B).$$

III. BAYESIAN NETWORK FOR IMAGE CLASSIFICATION

In order to tackle the weak assumption of conditional independence in NBNN, we need to find out the dependences among or within local features. Searching for the optimal dependences in the image, however, is an NP problem [10]. Instead, we model the dependences within local features by a simple Bayesian network (Sec. II-C). In this section, we introduce how to build and use our simple Bayesian network with local features. We then propose our optimization, relation-based Bayesian network, for effectively and efficiently considering the conditional dependence (Sec. III-C).

A. Our Naive Bayesian Network Model

Given an image, descriptors can be computed with different scales or at different locations. For our goal of considering the dependency within features, we extract features for a region in two different scales. Specifically, given a region, we represent the region with a single dense SIFT descriptor as well as four dense SIFT descriptors from 2 by 2 subdivided regions of the region, as shown in Fig. 4. We call the feature with a higher scale a high-level feature, and features with the smaller scale low-level features.

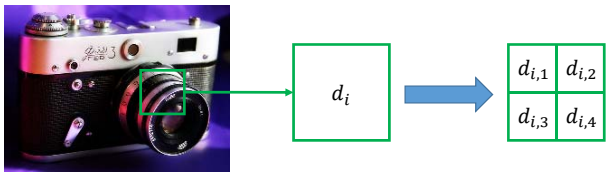

Fig. 4. For a image region, I_R, we extract a high-level feature, d_i, from the region, and four low-level features from 2 by 2 subdivided regions of the same region I_R. These features with two different scales represent the same region I_R.

We use d_i to denote the high-level feature of i-th image patch. d_i is a $n \times 1$ vector assuming that the dense SIFT has n dimensionality. We partition the i-th image patch into 2 by 2 sub-patches, and extract a low-level feature for each sub-patch that is also represented by an n dimensional dense SIFT. As a result, we have four different low-level features, $d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}$, from the same, i-th image patch.

We model our Bayesian network of local features in two layers. The first layer is constructed by the high-level feature, and is called a parent node. It is linked with the second layer. The second layer consists of a set of low-level features, which are named as children nodes. The layered structure makes a Bayesian network, and we use this structure of features for non-parametric classifiers. The left side of Fig. 5 shows our two-level layered Bayesian network.

B. Classification using Our Bayesian Network

We now explain how to utilize our Bayesian network, two-level layered structure of local features, to classify images. Before running the classification algorithm, we first prepare a set of labeled images for each class in a similar manner to the original NBNN. Unlike the original NBNN, the labeled image set for each class is represented by our layered image representation with their high-level and low-level features.

Given a query image Q, our method with the two-level layered Bayesian network assigns Q to one class based on the maximum likelihood estimation under the uniform prior among a possible class set indexed by C and the Bayes' rule. We then get $\hat{C} = \text{argmax}_c \log[P(Q|C)]$, as the original NBNN.

According to our two-level layered Bayesian network, we have the conditional dependence between the high-level and low-level features. Given a high-level feature d_i, it is associated with

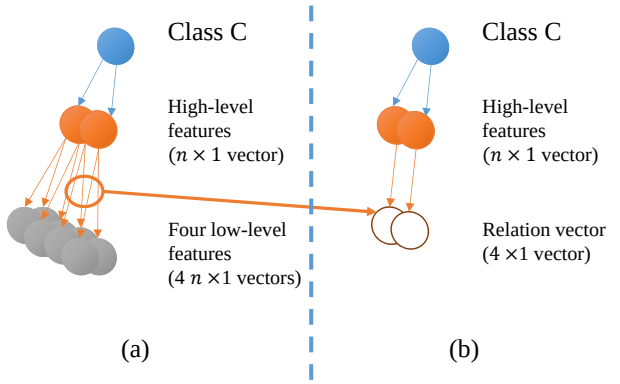

Fig. 5. (a) shows our naive two-level layered Bayesian network, and (b) shows our optimized representation, relation-based Bayesian network. (b) achieves high accuracy with low computational and memory overheads.

its four children nodes $d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}$. We then reformulate the aforementioned classification equation as the following:

$$\hat{C} = \underset{C}{\operatorname{argmax}} \log[P(d_i|C)P(d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}|d_i)]. \tag{7}$$

Under the parent node d_i, four children nodes $d_{i,1}, d_{i,2}, d_{i,3}, d_{i,4}$ are independent as the following:

$$P(d_{i,1}, \ldots, d_{i,4}|d_i) = \Pi_{j=1}^{4} P(d_{i,j}|d_i). \tag{8}$$

Additionally, an image can have n different high-level features, which are assumed to be independent. We therefore get the following classification:

$$\begin{aligned}\hat{C} &= \underset{C}{\operatorname{argmax}} \log \left[\prod_{i=1}^{n} P(d_i|C) \prod_{j=1}^{4} P(d_{i,j}|d_i)\right] \\ &= \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} (\log P(d_i|C) + \sum_{j=1}^{4} \log P(d_{i,j}|d_i))\right].\end{aligned} \tag{9}$$

In this equation we have two different conditional probability. We also use the Parzen window estimator with a Gaussian kernel K and approximate it with a nearest neighbor, $NN_C(d_i)$, from d_i in the class C, as adopted in the original NBNN. The conditional probability of descriptor d_i under the class C is then represented as follows:

$$\hat{P}(d_i|C) = \frac{1}{L} K(d_i - NN_C(d_i)), \tag{10}$$

where L is the number of high-level descriptors in the training (labeled) set for the class C. In a similar manner, we define the conditional probability of low-level descriptor $d_{i,j}$ under its parent node descriptor d_i as the following:

$$\hat{P}(d_{i,j}|d_i) = \frac{1}{l} K(d_{i,j} - NN_{d_t}(d_{i,j})), \tag{11}$$

Algorithm 1 Our two-level layered, Bayesian network for image classification.

1: Compute two-level layered descriptors of high-level descriptors $d_1, \ldots, d_n$ and their low-level descriptors $d_{1,1}, \ldots, d_{1,4}, d_{2,1} \ldots, d_{n,4}$
2: Compute the nearest neighbor (NN) of the descriptor
   (a) $\forall d_i\ \forall C$, compute the NN of d_i, $NN_C(d_i) \equiv d_t$, in C.
   (b) for each $d_{i,j}$ in d_i, compute the NN of $d_{i,j}$, $NN_{d_t}(d_{i,j})$, in the low-level descriptors of d_t.
3: $\hat{C} = \operatorname{argmin}_C[\sum_{i=1}^{n}(\|d_i - NN_C(d_i)\|^2 + \sum_{j=1}^{4} \|d_{i,j} - NN_{d_t}(d_{i,j})\|^2)]$

where l is the number of children nodes that the descriptor d_i has and set to be 4 in this paper. For evaluating the first conditional probability term, we already identify the nearest neighbor, $NN_C(d_i)$, of high-level descriptor d_i under the class C. In the labeled image set, $d_t \equiv NN_C(d_i)$ has four children nodes $d_{t,1}, \ldots, d_{t,4}$. As a result, $NN_{d_t}(d_{i,j})$ indicates the nearest neighbor of the low-level feature $d_{i,j}$ among low-level features $d_{t,1}, \ldots, d_{t,4}$ of d_t.

The kernel K is chosen as a Gaussian Kernel, which is substituted into Eq.10 and Eq.11. We then have the following classification function:

$$\begin{aligned}\hat{C} &= \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} (log \frac{1}{L} e^{-\frac{1}{2\alpha^2}\|d_i - NN_C(d_i)\|^2} + \right. \\ &\qquad \left. \sum_{j=1}^{4} log \frac{1}{l} e^{-\frac{1}{2\alpha^2}\|d_{i,j} - NN_{d_t}(d_{i,j})\|^2})\right] \\ &= \underset{C}{\operatorname{argmin}} \left[\sum_{i=1}^{n} (\|d_i - NN_C(d_i)\|^2 + \right. \\ &\qquad \left. \sum_{j=1}^{4} \|d_{i,j} - NN_{d_t}(d_{i,j})\|^2)\right].\end{aligned} \tag{12}$$

In summary, our two-level layered Bayesian network for image classification is shown at Algorithm 1.

Issues of the proposed Bayesian network. Our naive Bayesian network considers conditional dependence between high-level and low-level features. Its time and memory complexity, however, are higher than those of the original NBNN. Specifically, we need to perform four more nearest neighbor search to compute $NN_{d_t}(d_{i,j})$ for low-level features. While each search of these operations has only a small data, i.e., four low-level features for each high-level feature d_t, we need to perform a few thousands operations overall, since each image can have thousands of high-level features. Furthermore, storing those low-level features requires extra memory space. Additionally, low-level features extracted from a

region have location information. This spatial information, unfortunately, is not encoded in the classification function of Eq. 12.

C. Relation-based Bayesian Network

We extract a single high-level feature and its corresponding four low-level features from each image patch. As a result, they have intrinsic dependence or correlation between them. We therefore utilize such relationship for achieving high accuracy and even for lowering computational and memory overheads down.

To utilize such relationship, we propose to use a relationship vector, r. Specifically, each descriptor vector, d, for a high-level feature has a $n \times 1$ dimensionality, where n corresponds the dimensionality of the employed feature, e.g., 128 for the used dense SIFT. When we concatenate four low-level features into a matrix representation, these concatenated low-level features, denoted by d', live in a $n \times 4$ matrix space. We then represent the relationship between d and d' by r, which is a 4×1 vector, as the following:

$$r = {d'}^T d. \tag{13}$$

By applying the pseudoinverse, d^+, of d to the right side of each term in the above equation, and taking the transpose, we get the following equation:

$$d' = {d^+}^T r^T, \tag{14}$$

where $d^+ \equiv V\Sigma^+ U^*$ is computed by applying the singular value decomposition on d; V, Σ, U are 128 by 128, 128 by 1, and 1 by 1 matrices, respectively.

Given the equation (Eq. 14), a high-level feature, d_i, extracted from i-th patch, and its relationship vector, r_i, we can represent the concatenated four low-level features with the SVD counterpart of ${d_i}^T$, ${d_i^+}^T$, multiplied by r_i^T. The classification function shown in Eq. 12 can be then transformed as follows:

$$\hat{C} = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{n} (\|d_i - NN_c(d_i)\|^2 + \|{d_i^+}^T r_i^T - NN_{d_t}({d_i^+}^T r_i^T)\|^2). \tag{15}$$

Performing SVD for each high-level feature d_i is, however, a time-consuming process. Fortunately, when the first term has the low value, we can assume that d_i and $d_t = NN_C(d_i)$ are similar, and d_i^+ and d_t^+ are so accordingly. Based on this assumption, we simply drop d_i^+ and d_t^+ in the second term. Based on this simple approximation, we can avoid the expensive SVD computation. Furthermore, there is always a single relationship vector, r_t, associated with each high-level vector d_t and thus the nearest neighbor operation in the second term reduces to simply returning r_t. Since r_t is a 4 by 1 vector, we can drastically reduce the memory requirement.

Algorithm 2 Relation-based Bayesian network for image classification.

1: Compute high-level descriptors $d_1, \ldots, d_n$ and their relation vectors $r_1, \ldots, r_n$ with corresponding low-level descriptors.
2: Compute the nearest neighbor (NN) of the descriptor
   (a) $\forall d_i \ \forall C$, compute the NN of d_i, $NN_c(d_i) \equiv d_t$, in C.
   (b) Fetch the relation r_t of d_t in C
3: $\hat{C} = \operatorname{argmin}_C [\sum_{i=1}^{n} (\|d_i - NN_c(d_i)\|^2 + \lambda \|r_i - r_t\|^2)]$.

As a result, we have the following, efficient classification function:

$$\hat{C} \approx \underset{C}{\operatorname{argmin}} \left[\sum_{i=1}^{n} (\|d_i - NN_C(d_i)\|^2 + \lambda \|r_i - r_t\|^2) \right], \tag{16}$$

where λ is a weight factor for the second term. Note that our assumption breaks when two high-level features are similar, but their relationship vectors are significantly different. To mitigate this problem, while this problem occurs rarely in practice, we set λ value to be less than one. We have tried out different values within a range $(0, 1)$ and found 0.25 to show the best accuracy with our tested benchmarks.

As shown in the right side of Fig. 5, our relation based representation is quite simple. Instead of storing both high-level and low-level features, we compute the relation vector between them, and represent labeled image sets solely with high-level features and their associated relation vectors. This simple approach saves a lot of space and requires a minor computational overhead over the original NBNN, while considering conditional dependence among high-level and low-level features.

Our final relation-based Bayesian network for image classification is summarized in Algorithm 2.

IV. RESULTS AND DISCUSSIONS

We conducted a series of tests on the Intel Quadcore i7 3.60GHz with 16GB memory. We use the FLANN [19] library utilizing multiple, randomized kd-trees, to efficiently compute approximate nearest neighbors. In our experiments, we built the indexes of kd-trees with the labeled data set, and we load and use those pre-built data structures for performing nearest neighbor search in an efficient way.

Benchmarks We use the Caltech101 and Caltech256 benchmarks for various tests. Caltech101 contains 101 categories and about 40 to 800 images per category. Caltech256 has a set of 256 object categories, each of which has at least 80 images. By following the experiment protocol of the original NBNN,

Table I. Classification results of different methods with 20 test images in Caltech-101.

Method	Number of Labeled Images in Training Set	
	15	30
NBNN	0.4650	0.5206
LNBNN	0.4821	0.5620
BN	0.5016	0.5906
RBN	0.5971	0.7231

we randomly choose 15 (or more numbers of) images per category as labeled data for all the tests. For query images we randomly choose 20 test images from the benchmark except for already chosen training (labeled) images. We have iterated this process two times, and have measured the average precision for each class for accuracy comparisons.

Descriptor For each image we extract densely sampled SIFT descriptors [8]. The patch size of each high-level feature is set to be 32×32 pixels, and the patch is divided into 2 by 2 regions, each of where we extract a low-level patch. We extract high-level features at every 16 pixels along X and Y directions, and thus regions of high-level features have overlaps. If the size of an image is smaller than 200 we resize it to 200, and when the size is bigger than 450, we resize it to 450. In these configurations an image contains 300 to 2000 SIFT descriptors of high-level features.

A. Experimental Results

To show benefits of our method, we compare different types of our method against the other state-of-the-art NBNN methods, as the following:

- NBNN. The original NBNN method [3]
- LNBNN. The local NBNN method, a recently improved NBNN method [7]
- BN: Our two-level layered Bayesian network.
- RBN: Our relation-based Bayesian network that considers conditional dependence for the original NBNN.

We have implemented NBNN and LNBNN based on the guideline suggested by their corresponding papers.

Table I shows classification accuracy of different tested methods with 20 test images in Caltech101. All the methods achieve higher accuracy as we use more labeled data images, and our method achieves the highest accuracy. For the case tested with 30 labeled images for each class, our relation-based Bayesian network RBN shows 72.31%, which is more than 19% point higher than that of the original NBNN. Compared with LNBNN, our method achieves more than 16% point higher accuracy. This improvement is mainly thanks to considering the conditional dependency between high-level and low-level features.

Table II. Classification results by two methods LNBNN and RBN with 20 test images in Caltech-256

Method	Number of Labeled Images in Training Set			
	15	30	40	50
LNBNN	0.1535	0.1934	0.2424	0.2922
RBN	0.2982	0.4082	0.4879	0.5711

Table II shows the classification accuracy of LNBNN and RBN with 20 test images in Caltech256. We do not test NBNN and BN, since they are outperformed by LNBNN and RBN, respectively, as demonstrated in Caltech101. Compared with LNBNN, our method RBN shows higher accuracy over LBNN across all the tested setting. Especially with 50 labeled training data images for each class, our method achieves 57%, which is 27% point higher than that of LNBNN.

Fig. 6 shows the classification accuracy of different classes under 15 labeled training images and 20 test images. We can observe that in most cases, LNBNN performs better than NBNN, but shows worse results in some classes such as chair and cup. On the other hand, our method RBN shows better or similar accuracy over the tested prior method across different classes. This also demonstrates the robustness of our approach of considering conditional dependence among features.

In terms of runtime computational overheads, LNBNN is the fastest method over other methods including ours. Because the time complexity of our method is similar to NBNN, we gain the classification result for each query in the similar performance with NBNN. Our method RBN takes 2.5 s for a query image extracted by 1000 patches on average in Caltech101. We have tested adopting the idea of the local NBNN using a single, global kd-tree for all the classes. While it improves the runtime performance of our method, its accuracy was much lower than that of our RBN method. As a result, we maintain kd-trees, each of which is constructed from labeled images of each class, as proposed by the original NBNN.

V. CONCLUSION AND FUTURE WORK

We have proposed our two-level layered Bayesian network to consider conditional dependence among features for NBNN classification. Our method extracts a high-level feature and four low-level features for each image patch. We proposed our classification function based on nearest neighbors and our Bayesian network. To utilize spatial information between high-level and low-level features, and optimize the performance and memory requirement of our Bayesian network, we modify our classification into relation-based Bayesian network by using the 4×1 relationship vector between high-level and its corresponding four low-level features. We have

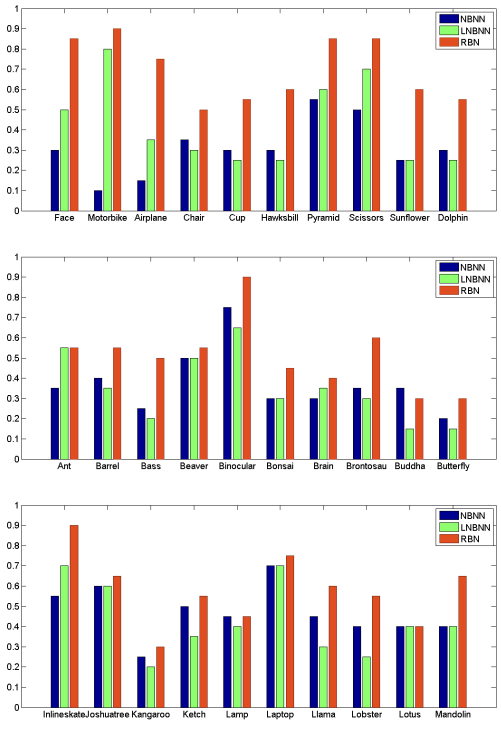

Fig. 6. Classification results by NBNN, LNBNN, and RBN with different classes. These results are acquired with 15 training (labeled) images for each class in Caltech101.

demonstrated benefits of our relation-based Bayesian network with two different benchmarks and showed that our method improves the accuracy over the original and local NBNNs. This result is achieved mainly thanks to considering conditional dependence between high-level and low-level features as well as encoding their correlation into the compact relationship vector.

As limitations and future work of our research direction, we would like to further utilize existing dependence among features. While we utilized conditional dependence between high-level and its low-level features, our method does not consider dependence among high-level features, as in the original NBNN method. We have tested our method mainly with Caltech101 and Caltech256, which are considered to be small compared to recent large-scale ones. To handle such recent ones for our method, we would like to adopt recent scalable nearest neighbor search techniques (e.g., hashing [20] and product quantization [21]).

Additionally, we followed all the guidelines of implementing NBNN and local NBNN techniques based on their corresponding papers. Results of our implementation of these techniques, however, showed lower accuracy over ones reported in other papers. We would like to refine our implementation so that we would like to bridge this gap. Nonetheless, our BN and RBN implementations share a lot of common parts (e.g., nearest neighbor search and dense SIFTs) with these NBNN methods. As a result, we also expect that our method can achieve higher accuracy, as we improve implementations of those NBNN techniques. Finally, we would like to improve the running performance of our method using multi-cores of CPUs and GPUs [22].

While our method has aforementioned limitations, we believe that our method takes a meaningful step for improving NBNN techniques by considering and encoding conditional dependency between high-level and low-level features.

ACKNOWLEDGEMENTS

This research was undertaken as part of MSIP/IITP [R0101-15-0176], the StarLab project of MSIP/IITP [R0126-15-1108], and NRF (2013R1A1A2058052).

BIOGRAPHIES

Mingyang Sun is currently at Mainframe Department of Agricultural Bank of China Data Center (ABCDC), ShangHai in China. Her job focuses on the IBM mainframe system operation and system-plex organization. In 2015, She received a M.S. degree in Computer Science Department from KAIST. Her research interest lies in image classification, image representation and bayesian network.

YoonSeok Lee is a M.S. student in School of Computing at KAIST, South Korea and he received a B.S. degree in Computer Science from KAIST in 2014. His research interest lies in image classification, image representation and hashing techniques.

Sung-Eui Yoon is currently an associate professor at KAIST. He received the B.S. and M.S. degrees in computer science from Seoul National University in 1999 and 2001, respectively. His main research interest is on designing scalable graphics, image search, and geometric algorithms. He gave numerous tutorials on proximity queries and large-scale rendering at various conferences including ACM SIGGRAPH and IEEE Visualization. Some of his work received a distinguished paper award at Pa-

cific Graphics, invitations to IEEE TVCG, an ACM student research competition award, and other domestic research-related awards. He is a senior member of IEEE, and a member of ACM and KIISE.

REFERENCES

[1] H. Zhang, A. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *CVPR*, vol. 2, 2006, pp. 2126 – 2136.

[2] E. Alpaydin, *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.

[3] O. Boiman and E. S. andM. Irani, "In defense of nearest neighbor based image classification," in *Proc. of Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[4] R. Behmo, P. Marcombes, A. Dalalyan, and V. Prinet, "Towards optimal naive bayes nearest neighbor," in *ECCV*, 2010.

[5] Z. Wang, S. Gao, and L.-T. Chia, "Learning class-to-image distance via large margin and l1-norm regularization," in *ECCV*, ser. Lecture Notes in Computer Science, 2012, pp. 230–244.

[6] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, "The nbnn kernel," in *ICCV*, 2011.

[7] S. McCann and D. G. Lowe, "Local naive bayes nearest neighbor for image classification," in *Proc. of Computer Vision and Pattern Recognition*, 2012, pp. 3650–3656.

[8] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *the International Conference on Computer Vision*, 2007, pp. 1–8.

[9] E. Bienenstock, S. Geman, and D. Potter, "Compositionality, mdl priors, and object recognition," *NIPS*, pp. 838–844, 1997.

[10] M. Sahami, "Learning limited dependence bayesian classifiers," in *Knowledge Discovery and Data Mining*, 1996, pp. 335–338.

[11] G. F. Cooper, "The computational complexity of probabilistic inference using bayesian belief networks," *Artificial intelligence*, vol. 42, no. 2, pp. 393–405, 1990.

[12] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, "The nbnn kernel," in *ICCV*, 2011.

[13] S. Vitaladevuni, P. Natarajan, S. Wu, X. Zhuang, R. Prasad, and P. Natarajan, "Scene image categorization and video event detection using naive bayes nearest neighbor," in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, 2013, pp. 140–147.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Proc. of International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] R. E. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, Inc. Upper Saddle River, 2003.

[16] D. D. Lewis, "Naive bayes at forty: The independence assumption in information retrival," in *Proc. of the 10th European Conference on Machine Learning*, 1998, pp. 4–15.

[17] D. Heckerman, "A tutorial on learning with bayesian networks," Tech. Rep., 1995.

[18] D. Geiger and T. Vermas, "Identifying independence in bayesian networks," *Proc. of Networks*, vol. 20, no. 5, pp. 507–534, 1990.

[19] M. Muja and D. Lowe, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *Proc. of Computer Vision and Pattern Recognition Workshop*, 2009, pp. 331–340.

[20] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, "Spherical hashing," in *CVPR*, 2012.

[21] J.-P. Heo, Z. Lin, and S.-E. Yoon, "Distance encoded product quantization," in *CVPR*, 2014, pp. 2139–2146.

[22] D. Kim, J. Lee, J. Lee, I. Shin, J. Kim, and S.-E. Yoon, "Scheduling in heterogeneous computing environments for proximity queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1513–1525, 2013.