Learning Discriminative Visual N-grams from Mid-level Image Features

Raj Kumar Gupta
gupta-rk@ihpc.a-star.edu.sg
Institute of High Performance Computing (A*STAR)
Singapore

Megha Pandey
pandeym@i2r.a-star.edu.sg
Institute of Infocomm Research (A*STAR)
Singapore

Alex YS Chia
alex.a.chia@rakuten.com
Rakuten Institute of Technology
Singapore

Abstract

The task of image classification is one of the key problems in computer vision, and has inspired a variety of image representations. In this paper, we propose a method to learn discriminative combinations of mid-level visual elements that capture their spatial configurations and co-occurrence relationships. We term such combinations as visual n-grams. Our method is capable of learning combinations with different number of elements. Experiments conducted on multiple datasets demonstrate the effectiveness of our approach where we achieve high image classification accuracy. Further, on fusing our features with global image features, we outperform the state-of-the-art results.

1 Introduction

Extraction of robust and informative representation is of crucial importance to image classification. Over the years, researchers have proposed a wide variety of image features. Bag-of-visual-words representations have been used widely and demonstrated successful classification performance with different kinds of low-level features [2, 19, 21]. They successfully capture low-level image information such as local edges and corners, but fail to provide much semantic information. Image-level features are able to capture substantial semantic information and have been shown to be effective for applications that involve finding near-duplicates from a tagged image dataset [10, 28]. These high level features are often learnt in a data-driven fashion thus necessitating the availability of huge datasets annotated with semantic tags.

Mid-level features [3, 17, 26, 27] can help bridge the semantic gap between pixel and image level representations. They are more informative, better understandable by humans and potentially more discriminative, as compared to low-level features. Also, they can be learnt in a bottom-up fashion without requiring a large annotated dataset. Many existing methods to learn mid-level visual elements consider each mid-level feature individually, and do not take their *mutual relationships* into account. We follow the intuitive idea that learning discriminative combinations of visual elements can help us deal with ambiguities better. As

Pages 47.1-47.12
DOI: https://dx.doi.org/10.5244/C.29.47

Figure 1: Example of a visual bigram. Similar visual elements may occur on different objects (highlighted in blue). Examining these elements together with neighboring patches (highlighted in yellow and red) can be useful in distinguishing the objects. Source images from [23].

an example, consider Figure 1 where a visually similar patch (in blue) can be found on both bike and motorbike objects. However, examining this patch along with those in the neighborhood provides better evidence towards distinguishing the two objects. Motivated by this, we propose the concept of visual n-grams to effectively represent combinations of visual elements along with their relative spatial configuration and co-occurrence relationships.

In this paper, we present a novel and effective method to automatically learn discriminative visual n-grams. Towards this end, we start with randomly extracted patches from the given set of training images, and employ categorical decision trees to learn a series of discriminative combinations. To successfully discover multiple discriminative n-grams, we incorporate a boosting framework and learn a series of categorical decision trees. We evaluate our method on the publicly available Graz-01 dataset [22], UIUC 8-sports events dataset [15], INRIA horse images dataset [1] and Land-Use dataset [31]. It is shown that our method attains high classification accuracy on these datasets and compares favorably with the existing methods. When fused with global image representation of Improved Fisher Vectors (IFV) [24], we outperform the state-of-the-art methods on these datasets.

2 Related Work

One form of mid-level representation can be obtained by pooling low-level features with the aim to retain more discriminative information than the standard bag-of-words representation does [3, 11, 14]. These approaches are able to perform better than the bag-of-words representation, but do not capture much of human understandable semantic concepts.

Yuan *et al*. [32] generate visual phrases as combinations of visual words based on their collocations patterns. Their representations, however, only includes co-occurrence statistics, while spatial layout information for co-occurring codewords is not considered. Chum and Matas [6] discover co-occurrences in high dimensional sparse data in an unsupervised manner. The co-occurrences learnt in this manner occur at a global level and are not necessarily effective in discriminating one class from another. We, on the other hand, explicitly learn discriminative combinations which take into account co-occurrence and relative spatial position and orientation information.

Representations based on Deformable Part Models (DPM) [8, 12, 23] learn object parts which have loose semantic connotations. They do well in capturing the frequently occurring structures in positive images. But they need to employ multi-component models to capture intra-class diversity, which is more computationally expensive. Our method is able to learn both frequently and infrequently appearing patterns without any additional computation. Additionally, each visual n-gram we learn represents a component of a scene as opposed to an entire object, and is able to detect those components even if the rest of the object is signifi-

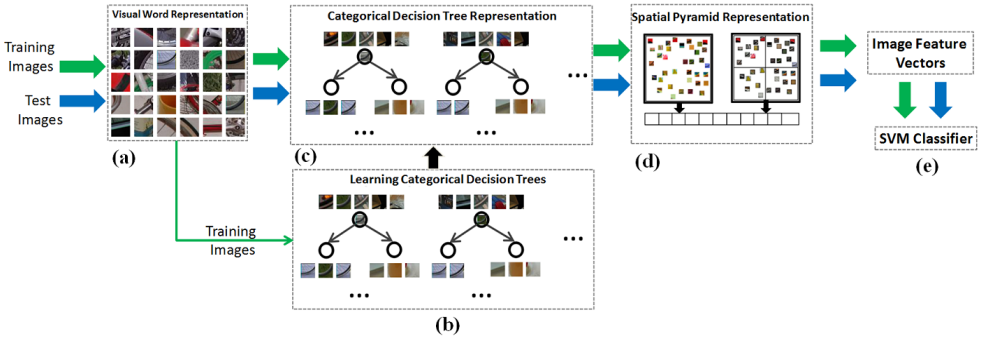

Figure 2: An overview of our approach. Our method (a) extracts mid-size patches at different scales and computes a visual codebook representation. (b) The mutual relationship between these visual words is learnt using categorical decision trees, which are then used to compute a feature vector for each image (c & d). (e) A classifier can then be learnt using standard SVM.

cantly occluded.

ObjectBank [16] and mid-level visual concepts [17] learn a set of object classifiers. The responses of these classifiers at different locations in an image are aggregated to obtain the overall image representation. These representations are not learnt in a bottom-up manner from the training data and are limited to a pre-defined finite list of visual concepts.

Singh *et al*. [26] proposed a method for unsupervised discovery of a set of discriminative mid-level patches appearing frequently in a dataset. They treat each patch individually and do not exploit the mutual relationships between them. They discussed the concept of *doublets*, i.e. pairs of mid-level patches that frequently co-occur in a certain spatial configuration. They, however, do not explicitly learn doublets. They are instead discovered after-wards based on the co-occurrence statistics of mid-level patches. Further, these doublets do not necessarily capture discriminative patterns. We, on the other hand, explicitly learn combinations of mid-level features that capture discriminative relationships. Our n-gram representation can learn combinations containing more than two patches as well.

3 Our Approach

Figure 2 gives an overview of our approach to learn discriminative visual n-grams. We start by densely extracting image patches from the training images, which are then each represented using a visual word descriptor. We adapt categorical decision tree model to represent and learn a combination of mid-level patches whose spatial co-occurrence relationship can provide a discriminative vote for the presence or absence of the target image class. We refer to such combinations as discriminative visual n-grams. To efficiently learn a diverse variety of such combinations, we employ a boosting framework. We exploit these n-grams along with a spatial pyramid framework to compute a feature representation for an image. The rest of this section discusses each of these steps in further detail.

3.1 Codebook Generation

First, we densely extract mid-size patches at different scales from the training images. We reject patches that have high overlap with other patches from the same image or have very weak gradient energy (and thus limited discriminative potential). A SIFT descriptor [19] is

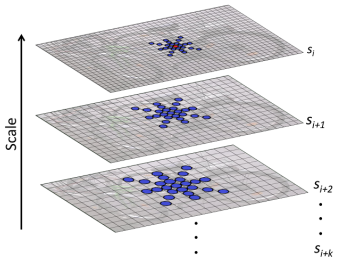

Figure 3: Spatial neighborhood for a given mid-level patch (shown in red). Patches included in the neighborhood are shown in blue. Best viewed in color.

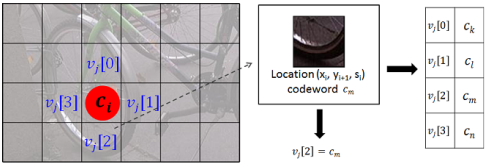

Figure 4: Spatial co-occurrence vector v_j extracted over a 4-neighborhood. Each dimension index refers to specific position and scale relative to the anchoring patch (in red). The value of each vector element captures the visual appearance of the respective neighboring patch.

computed for every patch. We apply standard k-means clustering to learn a codebook. The clusters that are either weakly populated or only contain members from the same training image are eliminated. The medoid of each of the clusters in this pruned set becomes a codeword. Once a codebook has been learnt, the SIFT descriptor for a given image patch can be quantized to the nearest codeword representation.

3.2 Learning a Discriminative Visual N-gram

Our goal is to next discover groups of codewords found to occur in each others' neighborhood in a given set of positive images, but not in negative images, or vice-versa. Given an image, we extract mid-level patches over a dense grid across multiple scales. Each patch is represented by the nearest codeword (in SIFT space). It is noteworthy that our framework is generic and can be used as it is with representations other than SIFT as well. For each patch, we extract a vector of the indices of codewords representing the patches in a spatial neighborhood. This vector implicitly encodes the spatial configuration information for the patch. A categorical decision tree model is employed to learn the combinations of codewords that co-occur in a set of images, and are able to vote for the presence or absence of an image category.

3.2.1 Spatial Co-occurrence Vector

The information about co-occurrence and relative positions and scales of different codewords is encoded by means of a spatial co-occurrence vector. For each mid-level patch in the training set, we define a neighborhood over nearby grid locations and adjacent scales. Multiple scales are included to obtain a representation robust to scale variation of the scene. A visual representation of this neighborhood is shown in Figure 3. The current patch is shown in red, and the blue dots denote the patches in its spatial neighborhood. For a grid location (x,y), we include patches within a 24-neighborhood of this location over k scales. We concatenate the indices of the codewords representing these neighborhood patches to obtain a $(24+25\times k)$ dimensional spatial co-occurrence vector. Each dimension index of this vector refers to a particular position and scale relative to the current patch, while the value of the corresponding vector element captures the visual appearance of this neighboring patch. A simple illustration can be seen in Figure 4. Capturing the spatial configuration in this manner empowers us to readily learn discriminative visual n-grams (details in Section 3.2.3).

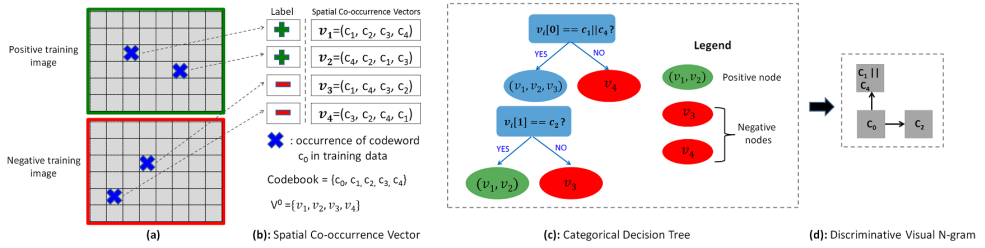

Figure 5: Learning a categorical decision tree. (a) All occurrences of codeword c_0 in training images. (b) Extraction of V^0, set of spatial co-occurrence vectors. (c) Categorical decision tree that best separates these vectors into positive and negative instances. (d) Visual n-gram.

3.2.2 Modeling a Visual N-gram

Let c_i and c_j denote codewords i and j. Given a mid-level patch at location (x,y) and scale s, we define an indicator variable $h_i(x,y,s)$ to denote the presence or absence of the codeword c_i. That is, $h_i(x,y,s) = 1$ if the the patch $p_{x,y,s}$ is represented by the codeword c_i, and 0 otherwise. Consider a combination of two codewords c_i and c_j appearing in locations (x_i,y_i,s_i) and (x_j,y_j,s_j) respectively. This combination can be mathematically represented as:

$$h_i(x_i,y_i,s_i) = 1, \quad and \quad h_j(x_j,y_j,s_j) = 1 \tag{1}$$

This representation can be further generalized as follows:

$$h_i(x_i,y_i,s_i) = b_i, \quad and \quad h_j(x_j,y_j,s_j) = b_j \tag{2}$$

where $b_i \in 0,1$ indicates the presence or absence of the codeword c_i. The above equation thus captures the co-occurrence relationship of two codewords at given locations.

Here we seek to learn combinations of codewords that capture discriminative information and can vote for the presence or absence of the targeted image class. A vote θ by a combination of codewords c_i and c_j can be represented as:

$$\theta(c_i,c_j) = \kappa \;\; if \quad h_i(x_i,y_i,s_i) = b_i, \;\; and \;\; h_j(x_j,y_j,s_j) = b_j \tag{3}$$

where $\kappa \in +1,-1$ denotes the presence or absence of the target image class. This equation represents a bigram, and can be further extended to n-grams including more than two codewords. A categorical decision tree provides a natural and intuitive choice to represent and learn these n-grams, where it can be shown that each path of the decision tree from the root node to the leaf node is mathematically equivalent to eqn. (3).

3.2.3 Categorical Decision Tree

We refer the interested reader to [25] for details on definition and representation of decision trees, and focus here instead on how we exploit them to discover discriminative visual n-grams. A categorical decision tree refers to a decision tree where the decision rules are not based on continuous variables, but on categorical variables.

An illustration of how we learn these decision trees is shown in Figure 5. We start by locating all appearances of codeword c_0 in the training images (Figure 5(a)). For each of these occurrences, we extract the corresponding spatial co-occurrence vector (Figure 5(b)). Let the set of all these vectors be V^0. Each of these vectors is assigned the same class

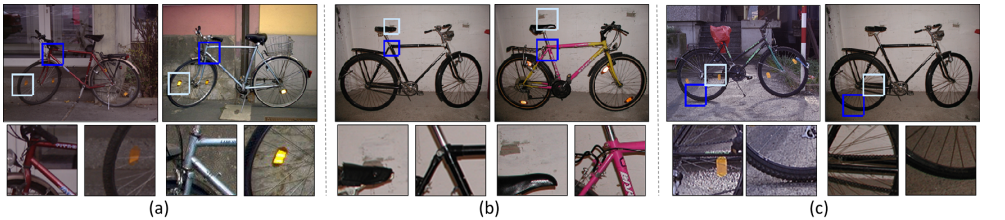

Figure 6: Examples of visual bigrams learnt on Graz-01 bike dataset. Each subfigure shows a bigram detection on two different images. Detected patches are highlighted in blue and white to show the relative spatial position. Comparisons of their visual appearance are shown in the bottom row.

label as that of the source training image. We now learn a decision tree that best splits V^0 into positive and negative instances (Figure 5(c)). To learn the decision rules, we employ a greedy approach that minimizes the classification error at all leaf nodes via maximization of information gain. Each path from the root node to a leaf node of this decision tree represents a visual n-gram, which can be mathematically represented as eqn. (3). Implicitly, this categorical decision tree captures the spatial configuration and co-occurrence relationships of a set of codewords that can vote for discriminating the positive and negative classes. As an example, the path from the root to the leftmost leaf node in the decision tree learnt in this illustration (Figure 5(c)) represents a discriminative visual n-gram (Figure 5(d)) which can vote for the presence of the positive image class as per the equation:

$$\begin{aligned} \theta(c_0,c_1,c_4,c_2) = 1 \quad if \quad & h_0(x_i,y_i,s_i) = 1, \\ & and\ h_2(x_{i+1},y_i,s_i) = 1 \\ & and\ (\, h_1(x_i,y_{i-1},s_i) = 1 \;||\; h_4(x_i,y_{i-1},s_i) = 1 \,) \end{aligned} \tag{4}$$

We refer to this decision tree and the corresponding n-grams as being anchored at the codeword c_0, since the decision rules consider values at locations defined relative to the location in which c_0 appears. In a similar manner, we learn a categorical decision tree anchored at c_i, for each codeword c_i in the codebook.

3.2.4 Boosting

The technique discussed above learns one categorical decision tree anchored at a codeword c_i. Such a representation, however, may not be comprehensive enough to handle the diversity in the image dataset. In particular, same codeword can potentially appear in different contexts in different images, or even different parts of the same image. To obtain a richer representation that can account for different spatial contexts that a codeword c_i might appear in, we employ a boosting framework to learn a series of categorical decision trees.

Figure 6 shows examples of visual bigrams learnt on Graz-01 *bike* dataset. Each bigram detects mid-level patches with similar appearance and same relative position on different images. The patches constituting the n-gram are highlighted in blue and white in the source images. The bottom row of each subfigure shows comparisons of their visual appearance.

3.3 Feature Computation

Having learnt a series of discriminative visual n-grams, they can now be used to compute a feature representation for an image. Given an image, we locate all the patches represented by the codeword c_i and extract the corresponding spatial co-occurrence vectors. We use the

Figure 7: Visual bigrams on *bike* dataset. These were detected on some positive as well as negative test images.

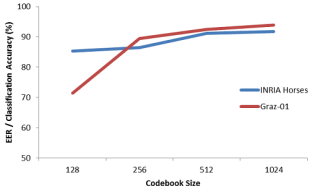

Figure 8: Classification performance over different codebook sizes.

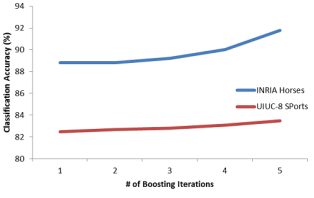

Figure 9: Classification rate over different number of boosting iterations.

categorical decision trees anchored at c_i to classify these vectors. Each path to a leaf node of a decision tree i.e. each visual n-gram contributes one feature value to the overall image representation. For a negative leaf node, the feature value is given by calculating the fraction of the vectors that are classified at this node, and is indicative of the occurrences of c_i that vote negatively for the target class. A higher feature value at such nodes will indicate lesser likelihood of the target class being present. A vector being classified at a positive leaf node implies that an n-gram belonging to the target class has been detected. Here, we compute the distance between the raw sift representations of image patches that constitute the n-gram and the codewords representing these patches, thus comparing the appearance of the detected occurrence with the visual n-gram we have learnt. To ensure the feature values are in the range 0 to 1, we apply the exponential function to the negative of the distance value. A higher feature value at these nodes implies that the detected combination closely matches the learnt visual n-gram, and thus votes strongly for the presence of the target class.

Feature values are computed in this manner for all the categorical decision trees we had learnt, and concatenated to obtain the complete feature vector. Motivated by the success of spatial pyramids [14, 23, 30], we further divide the image into four uniform quadrants and compute a feature vector for each. The concatenation of these four vectors and the image-level feature vector gives us the final image representation, which is L2 normalized to sum up to 1.

Finally, we employ the standard linear Support Vector Machines (SVM) to learn a one-versus-all model for each category. We use the libSVM [4] library.

4 Experiments

In this section, we evaluate the performance of our algorithm on the tasks of scene and event classification, and present quantitative and qualitative results. We test our approach on four publicly available datasets: Graz-01 [22], UIUC 8-sports event dataset [15], INRIA horse images dataset [1] and Land-Use dataset [31]. For all the experiments discussed here, we extract non-overlapping patches of size 8x8 pixels at 7 different resolutions. and learn a codebook of size 1024. The effect of varying codebook size on classification performance

Methods	Bike	Person	Average
Spatial Pyramid [14]	86.3	82.3	84.3
PIWAH + SPM [13]	87.4	84.6	86.0
NBNN [3]	90.0	87.0	88.5
HOF [8]	94.0	84.0	89.0
SPCK+ [31]	91.0	87.2	89.1
Singh *et al.* [26]	87.0	95.0	91.0
FLH + BOW [9]	**95.0**	90.1	92.6
GRID-FLH [9]	91.4	95.8	93.6
IFV [24]	93.0	96.0	94.5
Ours	93.0	95.0	94.0
Ours + IFV	94.0	**97.0**	**95.5**

Table 1: Equal Error Rates on Graz-01 dataset. All values in %.

Methods	Accuracy
ERC-Forest [20]	85.30
VQ [17]	91.40
VC [17]	92.47
IFV [24]	92.94 $\pm$ 0.62
Ours	91.76 $\pm$ 0.33
Ours + IFV	**94.71** $\pm$ **0.31**

Table 2: Classification accuracy on INRIA horse images dataset. All values in %. Standard deviations are computed over 10 random splits.

is shown in Figure 8. The performance in general increases as we increase the codebook size. We include $k = 2$ neighboring scales while defining the neighborhood to extract spatial co-occurrence vectors, resulting in a 74-dimensional vector. Using boosting we learn 5 categorical decision trees anchored at each codeword c_i, thus obtaining a total of 5120 categorical decision trees. Figure 9 shows a gradual but consistent upward trend in classification rate as we include more boosting iterations.

For fusing our image representation with IFV features, we concatenate the two feature representations. Before concatenation, we apply PCA and reduce the individual feature representations to 200 dimensions each. This helps limit the final feature vector to a low dimensionality, thus making the computations faster.

Below we discuss the results we obtained on the four datasets. Some qualitative results are shown in Figures 6, 7 and 10.

Graz-01 Dataset: This dataset consists of two object categories, *bike* and *person*, and a set of background images. In these images the targeted objects occur at different scales and poses, and are often placed in highly-textured background. Similar to the previous evaluations on this dataset [9, 22], we randomly sample 100 positive images and 100 negative images (50 each from the background set and the other object class) for both training and testing. We compare the ROC Equal Error Rate (EER) for our approach with other state-of-the-art methods in Table 1. We obtain an average EER of 94.0%, which compares favorably with the existing methods. Our representation offers an excellent complementarity to global image representations, such as Improved Fisher Vectors (IFV) [24]. Specifically, we fuse our mid-level feature representation with IFV, the result obtained from which is reported in the same table. We obtain an average classification accuracy of 95.5% which is higher than the result obtained using either feature individually, and is superior of those obtained by the state-of-the-art methods. Further, the mid-level representation in Singh *et al.* [26] achieves an EER of 91% compared to our 94%, thus showing that learning combinations is more discriminative than considering the mid-level patches individually.

Some qualitative examples of visual bigrams learnt on *bike* category are shown in Figure 6. They detect mid-level patches with same relative spatial configuration and similar appearance on different images. We show a few more examples in Figure 7 where the bigram is detected on some positive as well as negative test images. In Figures 7(a)&(b), the mis-classification occurs due to the presence of the same part (wheel) in both the images. In Figure 7(c), the detected combination does not belong to the same object part in the two images, but shows strong visual similarity. Another example on the *person* category is shown in Figure 10(a). This combination detects the face and neck of a person. In the middle image, a similar visual pattern appears on an advertisement poster where a match is found.

Methods	Accuracy
Object Bank [10]	76.30
Spatial Pyramid [4]	81.80
HIK [9]	84.21
Hybrid-Parts + GIST-color + SP [3]	87.20
VC + VQ [1]	88.40
IFV [4]	90.80 ± 0.12
ISPR + IFV [13]	92.08
Ours	83.54 ± 0.41
Ours + IFV	**93.12 ± 0.28**

Table 3: Average classification accuracy on UIUC 8-sports events dataset. All values in %. Standard deviations are computed over 10 random splits.

Methods	Accuracy
BOW [2]	71.90
Spatial Pyramid [17]	74.00
FLH + BOW [2]	77.20
SPCK++ [31]	77.38
GRID-FLH [2]	79.20
IFV [24]	85.05
Ours	79.52
Ours + IFV	**87.24**

Table 4: Average classification accuracy on Land-Use dataset. All values in %.

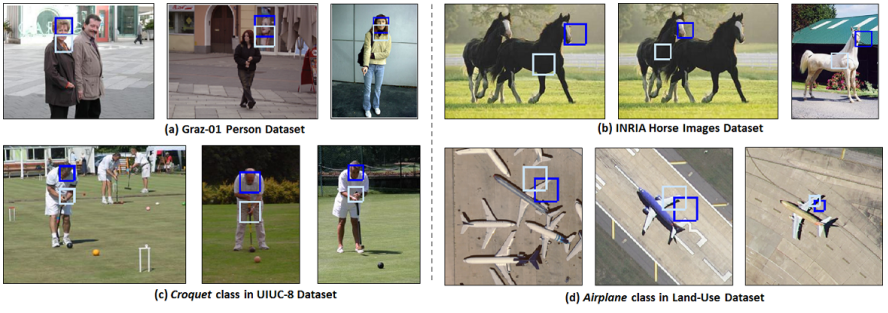

Figure 10: Examples of discriminative visual n-grams discovered on different datasets.

INRIA Horses Dataset: This dataset consists of 170 horse images and 170 background images. The dataset poses a challenge due to intra-class variability in shape and scale of the target object, as well as background clutter. Following [17], we randomly divide images from each class into two halves, and use one half for training and the other for testing. The experimental results are shown in Table 2. We obtain a classification rate that is comparable to the state-of-the-art. Together with IFV features, we obtain a classification accuracy higher than that given by either of the features individually, and outperform the best published results so far. Figure 10(b) shows an example visual n-gram that we learnt for this category.

UIUC 8 Sports Events Dataset: This dataset contains images from 8 sport events. Similar to [13], we randomly select 70 images for training and 60 for testing from each of the classes. Table 3 shows a comparison of the average classification accuracy we obtained with other methods. Here too, the combination of our features with IFV boosts the accuracy and outperforms all the state-of-the-art methods, again demonstrating the complementary nature of the two. Per-class classification accuracy obtained using the fused features is listed in the Table 5. We note that the performance is relatively weaker on the class *bocce* since both *bocce* and *croquet* are played in similar locations and hence share a lot of common visual features. Owing to this, 10% of *bocce* images in our test set were misclassified as *croquet*. In Figure 10(c), we show examples of a visual n-gram that detects some details of a player's posture in *croquet* images .

Land-Use Dataset: This dataset consists of 21 land-use classes extracted from aerial orthoimagery. Each class consists of 100 images measuring 256x256 pixels. We randomly split each class into two sets of 50 images, and use these for training and testing respectively. We compare the average classification accuracy from our experiments on this dataset with other state-of-the-art methods in the Table 4. We achieve higher performance than most other methods. In combination with IFV features, we again obtain improved performance

	badminton	bocce	croquet	polo	climbing	rowing	sailing	snowboarding
badminton	**98**	0	0	0	0	2	0	0
bocce	5	**75**	10	3	0	2	2	3
croquet	0	7	**92**	0	0	0	0	1
polo	0	0	0	**95**	2	2	1	0
rock climbing	0	0	0	0	**100**	0	0	0
rowing	2	0	0	0	3	**95**	0	0
sailing	0	0	0	0	0	2	**98**	0
snowboarding	0	0	0	0	5	3	0	**92**

Table 5: Confusion matrix (rounded values in %) obtained using our method combined with IFV [24] on UIUC 8-sport events dataset.

over both the features and outperform the state-of-the-art results. Examples of a visual n-gram detected on *airplane* are shown in Figure 10(d). It could successfully detect similar visual elements despite the object appearing at different scales in these images.

5 Conclusion

In this paper, we proposed an approach to learn discriminative combinations of mid-level patches. Such combinations, termed as visual n-grams, represent spatial configurations and co-occurrence relationships of mid-level visual elements that can best discriminate the target class from other classes. We exploit categorical decision trees to capture such relationships. Our algorithm is by nature flexible to automatically learn a variety of combinations with different number of visual elements and different configurations. Qualitative evaluation shows that our method can discover combinations of image parts that meaningfully capture details about the structural components of the scene being classified. We demonstrate the effectiveness of our image representation by applying it to the task of image classification on four datasets. Our method achieved high classification accuracy on each of these datasets, and by fusing visual n-grams based representation with global IFV features we achieved improved performance over the best published results so far.

References

[1] http://groups.inf.ed.ac.uk/calvin/inria-horses-v103.tgz.

[2] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. *Computer Vision and Pattern Recognition*, 2008.

[3] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. *Computer Vision and Pattern Recognition*, 2010.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[5] Tsuhan Chen and Yimeng Zhang. Efficient kernels for identifying unbounded-order spatial features. *Computer Vision and Pattern Recognition*, 2009.

[6] Ondrej Chum and Jiri Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *Computer Vision & Pattern Recognition*, 2010.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, 2005.

[8] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence*, 2010.

[9] Basura Fernando, Elisa Fromont, and Tinne Tuytelaars. Mining mid-level features for image classification. *International Journal of Computer Vision*, 2014.

[10] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. *Computer Vision and Pattern Recognition*, 2008.

[11] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision & Pattern Recognition*, 2010.

[12] Mayank Juneja, Andrea Vedaldi, C. V. Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. *International Conference on Computer Vision*, 2013.

[13] Rahat Khan, Cécile Barat, Damien Muselet, and Christophe Ducottet. Spatial orientations of visual word pairs to improve bag-of-visual-words model. In *British Machine Vision Conference*, 2012.

[14] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Computer Vision and Pattern Recognition*, 2006.

[15] Li-Jia Li and Li Fei-Fei. What, where and who? classifying events by scene and object recognition. *International Conference on Computer Vision*, 2007.

[16] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. *Neural Information Processing Systems*, 2010.

[17] Quannan Li, Jiajun Wu, and Zhuowen Tu. Harvesting mid-level visual concepts from large-scale internet images. *Computer Vision and Pattern Recognition*, 2013.

[18] Di Lin, Cewu Lu, Renjie Liao, and Jiaya Jia. Learning important spatial pooling regions for scene classification. *Computer Vision and Pattern Recognition*, 2014.

[19] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[20] Frank Moosmann, Eric Nowak, and Frederic Jurie. Randomized clustering forests for image classification. *Pattern Analysis and Machine Intelligence*, 2008.

[21] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence*, 2002.

[22] Andreas Opelt, Michael Fussenegger, Axel Pinz, and Peter Auer. Weak hypotheses and boosting for generic object detection and recognition. *European Conference on Computer Vision*, 2004.

[23] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. *International Conference on Computer Vision*, 2011.

[24] Florent Perronnin, Jorge SÃąnchez, and Yan Liu. Large-scale image retrieval with compressed fisher vectors. *Computer Vision and Pattern Recognition*, 2010.

[25] J Ross Quinlan. Induction of decision trees. *Machine Learning*, 1986.

[26] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. *European Conference on Computer Vision*, 2012.

[27] Jian Sun and Jean Ponce. Learning discriminative part detectors for image classification and cosegmentation. *International Conference on Computer Vision*, 2013.

[28] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: a large database for non-parametric object and scene recognition. *Pattern Analysis and Machine Intelligence*, 2008.

[29] Jianxin Wu and James M Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. *International Conference on Computer Vision*, 2009.

[30] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. *Computer Vision and Pattern Recognition*, 2009.

[31] Yi Yang and Shawn Newsam. Spatial pyramid co-occurrence for image classification. *International Conference on Computer Vision*, 2011.

[32] Junsong Yuan, Ying Wu, and Ming Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Computer Vision & Pattern Recognition*, 2007.

[33] Yingbin Zheng, Yu-Gang Jiang, and Xiangyang Xue. Learning hybrid part filters for scene recognition. In *European Conference on Computer Vision*, 2012.