Neti Neti: In Search of Deity

Yashaswi Verma
yashaswi.verma@research.iiit.ac.in

C. V. Jawahar
jawahar@iiit.ac.in

Center for Visual Information Technology, IIIT-Hyderabad, India - 500032

ABSTRACT

A wide category of objects and scenes can be effectively searched and classified using the modern descriptors and classifiers. With the performance on many popular categories becoming satisfactory, we explore into the issues associated with much harder recognition problems.

We address the problem of searching specific images in Indian stone-carvings and sculptures in an unsupervised setup. For this, we introduce a new dataset of 524 images containing sculptures and carvings of eight different Indian deities and three other subjects popular in the Indian scenario. We perform a thorough analysis to investigate various challenges associated with this task. A new image-representation is proposed using a sequence of discriminative patches mined in an unsupervised manner. For each image, these patches are identified based on their ability to distinguish the given image from the image most dissimilar to it. Then a rejection-based re-ranking scheme is formulated based on both similarity as well as dissimilarity between two images. This new scheme is experimentally compared with two baselines using state-of-the-art descriptors on the proposed dataset. Empirical evaluations demonstrate that our proposed method of image-representation and rejection cascade improves the retrieval performance on this hard problem as compared to the baseline descriptors.

1. INTRODUCTION

The traditional image descriptors based on bag-of-words BOW [21] and spatial pyramids [15] have emerged as successful baseline solutions for most of the modern recognition and retrieval tasks, such as instance or category-based retrieval [3, 8, 21] and classification [5]. These descriptors are built using illumination/scale/view-invariant features such as SIFT [16] extracted at (local) interest points or on a dense grid. However, these are mostly useful in searching specific/identical textured objects, but are usually not very good for large category of varied objects. In such scenarios, GIST [17]; which is a continuous global descriptor; is found to perform better [7]. Recently, there has been focus on developing new descriptors that try to address some of the limitations of interest-point based features. E.g. in [1] a new representation; called "bag of boundaries" (or BOB); was proposed for retrieving smooth objects with fixed geometry and minor variations in orientations.

Figure 1: Example images from our dataset. Note the variations in colour, texture, view-point, lighting, shape, size, appearance and posture.

Along with these developments, significant efforts have been put into developing new models that complement the modern descriptors, and are capable of modelling the shape and relative position of the parts of objects [10]. These models are object-specific and have been found to perform well for detecting categories such as car, person, etc. allowing little variations in pose. However, as noted in [18], for objects that have highly flexible and deformable parts (e.g. cats and dogs), simple BoW models outperform the state-of-the-art Deformable Part Model (DPM) [9].

In this paper, we address the problem of unsupervised retrieval of carving images similar to a given query image. In such images, either there is a carved sculpture, or a carving made on a background of the same material (mostly coloured stone). Such carvings can easily be found in and around ancient caves and temples, and are an indispensable part of Indian heritage. Previous works focussing on sculpture retrieval [1, 2] have tried to address the problem of instance-based retrieval on a collection of symbolic shapes (e.g. *"The Thinker"* [1]). Such shapes are quite often reproduced at dif-

[1] `http://en.wikipedia.org/wiki/The_Thinker`

ICVGIP '12, December 16-19, 2012, Mumbai, India

Figure 2: Different carvings of *Ganesh* illustrating variations in posture, number of hands and gadgets.

ferent places; and the major challenges involved in retrieval are size, view-point, and surface-properties (textured/non-textured). In our case, retrieval becomes further challenging because the sculptures/carvings of even a single deity can vary a lot. E.g., as shown in Figure 2, *Ganesh* can have different postures, number of hands and additional gadgets. This poses an additional question that whether the retrieval of Indian sculptures is instance-based or category-based. Also, there is no useful context information available which is the case with the well-known PASCAL dataset (it was also observed in [1] that background information improves retrieval performance). To address these challenges, we formulate the retrieval task as an unsupervised rejection-based scheme that incorporates both similarity as well as dissimilarity between a pair of images. This scheme is inspired from the concept of *Neti-Neti* (see Sec. 6).

In this paper, we introduce a new dataset of Indian stone carvings. It contains images from eleven different categories, including eight Indian deities (*Buddha*, *Durga*, *Ganesh*, *Hanuman*, *Krishna*, *Nandi*, *Natraj* and *Saraswati*) and three other subjects popular in Indian carvings (*Elephant*, *Horse* and *Wheel*). Figure 1 shows few example images from the dataset. As is evident from these images, it is not straightforward to perform retrieval over these because

(i) Each carving is unique irrespective of its category, with significant variations in shape, size, appearance, orientation and even material. In addition, the carvings can have smooth as well as edgy surfaces, and even different carvings can have similar edges. This makes it difficult to distinguish them using interest point (BOW using SIFT) or boundary (BOB) based descriptors.

(ii) Though there exist one or more distinctive parts for most of the categories, they themselves might be quite flexible, articulated and even occluded/eroded due to which locating them using DPM [10] becomes non-trivial.

(iii) Since the carvings are mostly made on a piece of stone, a significant region surrounding the object of interest has similar colour and texture properties as that of the object itself; hence super-pixel based foreground-background segmentation (as used in [1, 4] for retrieval) is seldom accurate.

These challenges make it difficult to either directly apply a part-based model or use existing descriptors for this task. To this end, we propose a new representation for the carving images. To minimize the conflict with background, first the given image is automatically segmented into foreground (carving) and background. The foreground is represented using a sequence of patches that try to capture the properties of discriminative regions in a carving. Since these patches both represent an image as well as capture characteristic local discriminative regions specific to a carving, these can be thought of as a combination of the part-based model and interest-point based descriptors. We use this representation in a re-ranking framework. For a given query image, first the dataset images are ranked using a simple baseline descriptor (such as SIFT or GIST), and then only the top few results are re-ranked using the patch-based representation. To validate the effectiveness of this new representation, we compare it with two baselines on the proposed dataset. Experimental evaluations demonstrate that our proposed scheme (i.e., image-representation + re-ranking) improves the retrieval performance as compared to the baselines, and hence is capable of addressing some of the challenges involved in this task.

2. DATASET AND CHALLENGES

Our dataset contains 524 images from eleven different categories, with around $45 - 50$ images per category. We denote the i^{th} category by C_i for $i = 1, \ldots, 11$. In this section, we analyze the inter-class and intra-class variabilities in our dataset (similar analysis was done in [7] for the ImageNet dataset). This will help in developing insights about the dataset and will also justify the complexity of the problem being addressed. For this, we use two different descriptors: GIST and BOW histogram using SIFT. These two descriptors are extensively used and have been shown to achieve promising results for classification and retrieval tasks [5, 8]. For GIST, we use the default parameters (number of orientations per scale $= [8, 8, 8, 8]$ and number of blocks $= 4$) resulting in a feature vector of 512 dimensions. For BOW, we use the SIFT descriptors computed densely on a regular image-grid with spacing of 3 pixels and quantize them into a vocabulary of 1000 visual words using the k-means algorithm. For computing the GIST features, we use the code from [17]; and for the SIFT features, we use the VLFeat library [22]. Both of these descriptors are L_2 normalized and Euclidean measure is used for computing distance between two features.

(i) Visual Scale

This term measures the visual variability within the samples of a category. For each category C_i, we compute its visual scale as the average distance of its mean descriptor from all the images in C_i. Another way of analyzing the visual scale of a category is by first determining a prototype image for that category, and then finding how well it approximates all other images within that category (based on average distance from all other images). For a given category, its prototype image is the one whose average distance from all other images in that category is minimum.

(ii) Semantic Separation

This term measures how well-separated the different categories are from each other. For this, we compute average distance of the mean descriptor of a category C_i to the mean descriptors of all other categories.

Figure 3 demonstrates the above statistics for the eleven categories of our dataset computed using the two descriptors discussed above. Though it was shown in [7] for the ImageNet dataset that modern descriptors are capable of separating semantically meaningful classes, that conclusion does not really apply to our dataset. Figure 3 shows that unlike the ImageNet dataset, our dataset is highly heterogeneous and there is little visual similarity among the images within a class. Interestingly, for all the categories (except the first one in case of GIST), the average distance of the

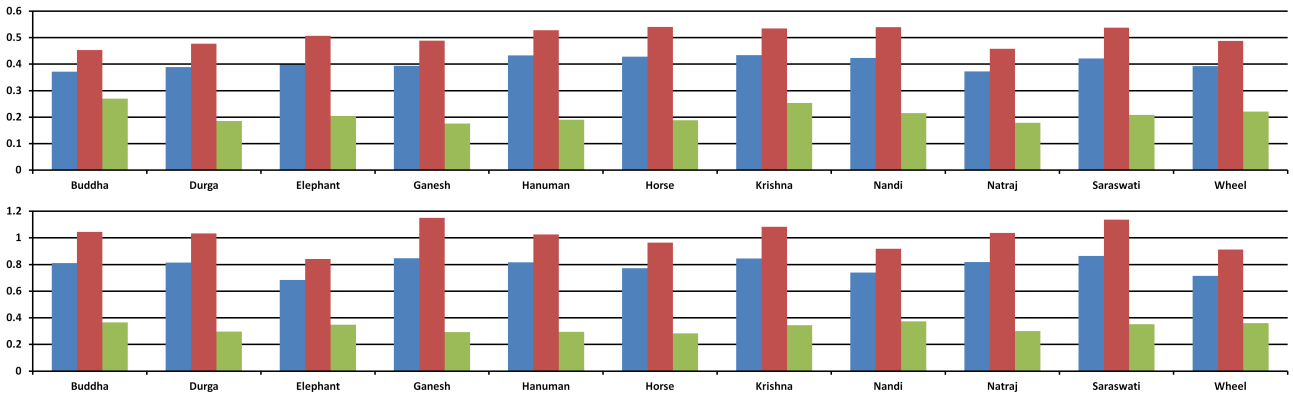


Figure 3: Different statistics for our dataset. The first bar (blue) shows the average distance of the mean descriptor of a category from all the images in that category. The second bar (red) shows the average distance of the prototype image of a category from all other images in that category. And, the third bar (green) shows the average distance of the mean descriptor of a category from the mean descriptors of all other categories. The statistics in the top figure are computed using the GIST descriptor, and those in the bottom figure are computed using BOW with the SIFT descriptor. See Sec. 2 for details.

mean descriptor of a class from the mean descriptors of all other classes (semantic separation) is nearly half or less than half of its visual scale (intra-class variability). This means that on an average, intra-class variabilities are far more than inter-class variabilities. This analysis demonstrates incapability of modern descriptors in separating the semantic classes of our dataset, and also verifies problem complexity. [2]

3. IMAGE REPRESENTATION

We represent a carving using a sequence of local discriminative patches. As it was shown previously in [1, 4] that segmentation of the object of interest from the surrounding significantly improves the recognition accuracies, we first segment the carving region from the background, and then compute its representation.

3.1 Foreground Segmentation

In object classification problems such as classification on the PASCAL dataset [9], context knowledge (*water*) can be very helpful in identifying an object (*ship vs. car*). However, previous works have demonstrated the need for segmenting the object of interest for addressing harder problems of fine-grain or within-class classification (e.g. flower classification [4]). In such scenarios, the context information is either unavaiable, or not useful (all flowers have the background of leaves). Our problem is also along the similar lines, hence we perform foreground segmentation as a pre-processing step. In our case, the objects of interest are either carvings or sculptures that are mostly made of stone. Though it might be straightforward to use a supervised segmentation approach based on a classifier trained on possible foreground/background regions [1, 4, 18], we propose an unsupervised method for this. This is because there are large variations possible in the surface properties of possible foreground regions. This makes learning a good region-classifier difficult, and hence makes supervised segmentation unsuitable for this task. For segmentation, we assume that the object of interest is also visually most salient [14]; or, there is some region contained in the object that is visually most salient as compared to all other regions. Then, the segmentation is performed as discussed below:

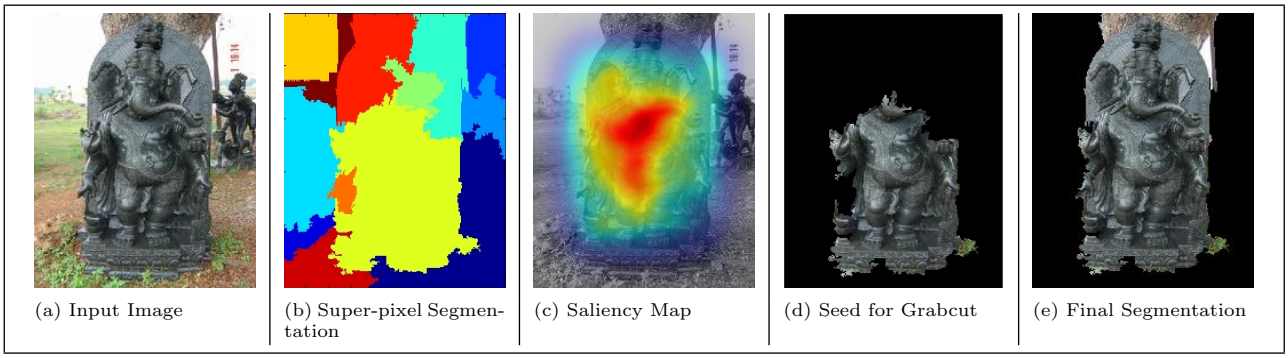
(i) Super-pixel Segmentation

We are interested in generating a coarse segmentation of the carving in an image. This is because many a times, the carvings are made on a stone wall which has similar colour and texture properties; hence it might not be possible to perfectly segment it out using some modern segmentation algorithm. Given an image, we segment it into super-pixels using the code from [11]. This is a greedy graph-based segmentation method that uses the boundary information between a pair of regions. To perform segmentation, we set $\sigma = 0.4$, $K = 500$ and $minimum\_region\_size = 500$ $pixels$. This gives approximately 15 super-pixels per image on an average. We use quite a large value for the "minimum region size" so that we get large super-pixels. This is particularly useful for our task because, as discussed above, we want coarse foreground segmentation. It is admissible to get some background region in the finally segmented foreground, but we do not want to miss any region that should actually come in foreground, thus emphasizing high recall.

(ii) Seed-selection and Grabcut

Visual-saliency plays an important role in determining the region(s) of interest in an image/object [20]. The visual-saliency of an object's region also hints about how informative it is in that object's identity. Hence it can be used as a measure for determining the most informative region(s) in an object. Inspired from this, we use a graph-based visual saliency method [14] for selecting the most salient region from the regions obtained after super-pixel segmentation. It is a bottom-up model that gives a visual saliency score in the range $[0, 1]$ to each pixel, with higher score corresponding to higher visual saliency. We use the super-pixel with largest

[2]We also experimented with other descriptors such as HOG and LBP using BOW histogram, but they were found to perform worse than SIFT.



(a) Input Image (b) Super-pixel Segmentation (c) Saliency Map (d) Seed for Grabcut (e) Final Segmentation

Figure 4: Pipeline for unsupervised object-of-interest segmentation. (a) Given an image, (b) first we segment it into super-pixels and (c) find the most salient super-pixel. (d) This super-pixel is then used as seed for initiating Grabcut to perform foreground segmentation in an iterative manner, which finally returns the segmented foreground (e).

cumulative saliency score as foreground seed for initiating Grabcut [19], and rest of the image pixels are considered as background. Keeping the size of super-pixels large practically helps in expanding the initial seed, thus preventing the Grabcut algorithm from returning the complete image as background. We perform Grabcut using the code from [13]. The output of Grabcut is uniformly expanded by a small fraction and again used as seed for updating the foreground in an iterative manner. After each iteration, if more than half of a super-pixel comes into foreground, then that super-pixel is completely merged into the foreground. We perform 25 iterations per image for foreground segmentation.

Figure 4 illustrates our foreground segmentation steps. It is important to note that our segmentation pipeline is completely unsupervised unlike previous approaches [1, 4, 18]. Figure 5 shows some results of image segmentation. It can be seen that our segmentation approach performs reasonably well and is capable of segmenting the object of interest.

(iii) Addressing Segmentation Failures

Because of the challenges involved in segmentation, it is not always possible to get a reasonably segmented foreground. In our case, since a significant portion of image is occupied by either the sculpture or the same material as that of the sculpture, we assume that at least one-third of image pixels should be classified as foreground in the segmentation output. In cases where this does not hold, we discard the segmentation output and pick the foreground pixels just based on the saliency score (pixels with saliency ≥ 0.25).

3.2 Patch Descriptor

Due to large intra-class and small inter-class variabilities, the traditional image descriptors such as BOW using SIFT might not be suitable for image-representation. However, it is possible to determine some patches in a carving that are most discriminative with respect to that particular carving (similar idea was adopted in [12], though for a different task). Following this, we propose a method for representing an image using a fixed number of ordered patches. These patches are identified such that they are most discriminative following some criteria (that we will discuss below). Below we discuss our method for image representation.

(i) Determining Different-Class Pairs

Let $\mathcal{T} = \{I_1, \ldots, I_t\}$ be the set of images. For an image I, we identify its farthest neighbour J from $\mathcal{T}$ using a standard descriptor (say GIST). Since these two samples are farthest from each other, they are very likely to be from different classes. We perform this for every image and form a set $\mathcal{T}_{neg}$ consisting of all such pairs of images.

(ii) Identifying Discriminative Patches

For each image I, we now have another image J that is most discriminative to it. We represent both these images using a collection of square patches around interest points described using the Histogram of Oriented Gradients (HOG) descriptor [6]. We prefer the HOG descriptor over other descriptors because it can capture the local shape and appearance characteristics of an object and is more beneficial in our case. Also, using the HOG descriptor, we get a representation for each patch in the continuous space. Instead, if we had used the SIFT descriptor, each patch would have been represented using a histogram of discrete BOW that would suffer from the quantization loss.

The patches are computed by resizing an image $\{0.5, 0.75, 1.0, 1.25\}$ times the original size. Each patch is a block of 2×2 HOG cells and each cell contains 8×8 pixels. For each cell, a HOG descriptor is computed using the method and code from [10]. For each cell, this gives a 31 dimensional feature vector, which are then concatenated to obtain a 124 dimensional HOG descriptor for each block (or patch). This descriptor is L_2 normalized for comparing two patches using the Euclidean distance.

Let $P_I = \{p_1, \ldots, p_{n_I}\}$ and $Q_J = \{q_1, \ldots, q_{n_J}\}$ be the set of patches corresponding to image I and J respectively. Our aim is to pick a *sequence* of m most discriminative patches $(m \leq min(n_I, n_J))$ $F_I^m = \{f_I^1, \ldots, f_I^m\}$ from the set P_I. We perform this in an iterative manner using a greedy approach. In the beginning, we select the first patch f_I^1 from P_I such that its maximum distance from any patch in Q_J is more than the maximum distance of any other patch in P_I from the pathces in Q_J. Precisely,

$$f_I^1 = \arg\max_i \max_j ||p_i - q_j||^2, \quad (1)$$

where the index i and j vary over patches in P_I and Q_J respectively. Now, suppose we have picked k patches $F_I^k =$

Figure 5: Examples of unsupervised foreground-segmentation. The first row shows example images from the dataset, and the second row shows corresponding segmented foregrounds.

$\{f_I^1, \ldots, f_I^k\}$. To pick the next patch f_I^{k+1} $(k+1 \leq m)$, we use the same procedure as above, leaving out the previously picked patches. After m such iterations, we will get a sequence of m patches F_I^m corresponding to I. This sequece will be such that the patches with lower indices will be more discriminative than the patches with higher indices, because of the way they are seleted. In practice, we pick $m = 20$ patches per image. It is important to note that in this step, we are also rejecting a large number of patches which are not very discriminative. This helps in reducing the size of image representation, and at the same time provides a representation in the continuous space.

Though our approach of identifying discriminative patches is inspired from [12], we have formulated a completely different procedure for doing so in the retrieval task, with the major difference being the complete unsupervised nature of our approach. We do so because in image retrieval scenarios, it is usually not desirable to harness category-level information of the dataset images.

4. PARAMETRIZING PATCHES

Image retrieval is usually performed by ranking all the dataset images given a query image using some measure such as Euclidean distance. The distances are computed in the feature space; hence rather than simply using the unweighted features, it is desirable to learn a metric that parametrizes the distance between two image features. In this section, we discuss a metric learning approach that learns two metrics using dissimilarity and similarity between pairs of images.

4.1 Parametrizing Different-Class Patches

For each image I, we have a sequence of m most discriminative patches $F_I^m = \{f_I^1, \ldots, f_I^m\}$. Recall that each of these patches is represented by a HOG descriptor. So, we can represent I using a single feature vector $\mathbf{x_I} \in \mathcal{R}^{\mathcal{N}}$ which is a concatenation of HOG descriptors of the patches $\{f_I^1, \ldots, f_I^m\}$ in that sequence. This way, we get a single feature vector representing an image. Now, from $\mathcal{T}_{neg}$, we have pairs of images such that the images in each pair are very likely to be from different classes. Let $(I, J) \in \mathcal{T}_{neg}$, and $\mathbf{x_I}$ and $\mathbf{x_J}$ be the feature representations of I and J respectively. We form a vector $\mathbf{d_{IJ}} \in \mathcal{R}_+^N$ such that $\mathbf{d_{IJ}}(k) = (\mathbf{x_I}(k) - \mathbf{x_J}(k))^2$, (where $(\cdot)$ denotes an entry in a vector). That is, each element of $\mathbf{d_{IJ}}$ denotes the squared Euclidean distance between the corresponding elements of $\mathbf{x_I}$ and $\mathbf{x_J}$. From this, we define the parametrized distance between the two images I and J as:

$$D_w(I, J) = \mathbf{w} \cdot \mathbf{d_{IJ}}, \tag{2}$$

where the vector $\mathbf{w} \in \mathcal{R}_+^N$ parametrizes the distance between two images that are likely to be from different classes. Our next task is to learn the metric $\mathbf{w}$. The traditional metric-learning algorithms such as [23] usually require samples or sample-pairs from same class as well as different class(es). Since all we have are the samples (most likely to be) from different classes, it might not be straightforward to estimate $\mathbf{w}$ using [23] or any other existing algorithm. Here, we formulate metric-learning as an optimization problem that suits our requirements; i.e. learns $\mathbf{w}$ just using the samples from $\mathcal{T}_{neg}$. Precisely, we are interested in minimizing the following objective function:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + \mathcal{C}_1 \textstyle\sum_{(I,J)\in\mathcal{T}_{neg}} [\alpha - \mathbf{w} \cdot \mathbf{d_{IJ}}]_+ \\ s.t. \quad & \mathbf{w}(i) \geq 0 \quad \forall i. \end{aligned} \tag{3}$$

Here, $\mathcal{C}_1 > 0$ handles the trade-off between the two terms, $[\,]_+$ is the standard hinge-loss such that $[z]_+ = max(0, z)$, and $\alpha > 0$ is the margin constraint. In our case, if we do not put any constraint on $\mathbf{w}$, then the objective function will go on minimizing by simply scaling it, without any learning. In order to control this, we regularize $\mathbf{w}$ which will prevent it from arbitrarily scaling. Also, we want every element of $\mathbf{w}$ to be non-negative so that Eq. 2 obeys the non-negativity property of a distance metric. Therefore, after each update, it is projected back into $\mathcal{R}_+^N$. After solving the above optimization, we obtain the metric $\mathbf{w}$ that computes *dissimilarity* between two images. Note that our metric learning approach is completely unsupervised unlike [23], i.e. we do not use category information anywhere; instead, we use the image pairs that are very likely to be from different classes.

4.2 Determining Same-Class Pairs

As we formed the set $\mathcal{T}_{neg}$, similarly we can form the set $\mathcal{T}_{pos}$ as well. For an image $I \in \mathcal{T}$, we identify its most similar image J from $\mathcal{T}$. But here, unlike the previous case, we can not very confidently claim whether both the images will be from the same class because of large intra-class varibilities (as discussed in Sec. 2). However, since we have learnt the vector $\mathbf{w}$, we can use this in rejecting *bad* pairs. For each pair of most similar images, we find their parametrized distance (Eq. 2), and pick only the top 50% pairs with least distance. This way, we extract the pairs which are more likely to belong to the same-class. Using all such pairs, we form the set $\mathcal{T}_{pos}$ analogues to $\mathcal{T}_{neg}$.

4.3 Parametrizing Same-Class Patches

Similar to Sec. 4.1, we can define distance between any pair of images $(I, J) \in \mathcal{T}_{pos}$. Precisely, $D_v(I, J) = \mathbf{v} \cdot \mathbf{d_{IJ}}$, where $\mathbf{v}$ is analogues to $\mathbf{w}$ except that it parametrizes similarity between two images. In this case also, we can optimize $\mathbf{v}$ in a manner similar to Eq. 3. However, the objective function will change because now we have to minimize the distance between images in a pair. We define the objective function as below:

$$\begin{aligned} \min_{\mathbf{v}} \quad & \tfrac{1}{2}||\mathbf{v} - \mathbf{1}||^2 + \mathcal{C}_2 \textstyle\sum_{(I,J)\in\mathcal{T}_{pos}} [\mathbf{v} \cdot \mathbf{d_{IJ}} - \beta]_+ \\ s.t. \quad & \mathbf{v}(i) \geq 0 \quad \forall i. \end{aligned} \tag{4}$$

Here, $\mathcal{C}_2 > 0$, $[\,]_+$ denotes hinge-loss, and $\beta > 0$ is the margin constraint ($\beta < \alpha$). In the above optimization, in order to prevent $\mathbf{v}$ from being a zero-vector, regulaization is performed that penalizes its variations from uniform map. One interesting thing to note here is that we have used different margins in the two optimization problems (Eq. 3 and Eq. 4). Intuitively, we want samples from the same class to be closer as compared to those from different classes by atleast a margin of $\alpha - \beta$. This way, we indirectly introduce a (weak) large-margin constraint analogues to [23] in an unsupervised setting.

One might say that the learnt weights $\mathbf{w}$ and $\mathbf{v}$ are not very strong metrics individually, because of the procedure and assumptions made during learning. However, they can be combined to provide a better similarity measure which we will discuss in the next section.

5. RETRIEVAL PROCEDURE

Here, first we would like to briefly summarize the previous steps. First, we segment the carving region of an image. Next, on the segmented region, we compute patch descriptors and then selectively identify a sequence of discriminative pathces. And finally, we learn metrics $\mathbf{w}$ and $\mathbf{v}$ using pairs of (most likely) different and same-class images respectively.

Given a query image Q, our goal is to retrieve dataset images such that most similar images are ranked higher as compared to rest of the images. For this purpose, first we rank all the images with respect to Q using a baseline descriptor (GIST or BOW with SIFT) and obtain the lowest ranked image (the image most dissimilar to it). Using the procedure discussed in Sec. 3.2, we identify the sequence of m discriminative patches in Q and concatenate them to form a feature vector $\mathbf{x_Q}$. After this, we determine the similarity of any image I in the dataset with Q using the following measure:

$$S_{I,Q} = \frac{(\mathbf{v} \cdot \mathbf{d_{IQ}})^{-1}}{\mathbf{w} \cdot \mathbf{d_{IQ}}}. \tag{5}$$

Here, $\mathbf{d_{IQ}}$ is the distance vector corresponding to the feature vectors $\mathbf{x_I}$ and $\mathbf{x_Q}$ of the two images (as discussed in Sec. 4.1). In the above score, we compute both how similar (numerator) and how dissimilar (denominator) are the given two images. This provides a stronger measure of comparing two images as compared to using a single metric, and is similar in idea to the log-likelihood ratio used in [12] and ratio-test in [16] for calculating relevance between two images. Using the above score function we can rank the dataset images based on their score, with larger score corresponding to higher relevance (i.e., ranking based on descending order of the score). However, since this is computationally intensive to calculate score for all the images, we use our method for re-ranking only the top few results retrieved using a baseline descriptor (such as SIFT or GIST). This is a sort of a rejection-based refining scheme where at first step, few most confident candidates are identified using a weak criteria, and then some stronger but computationally intensive criteria is used to refine previously obtained results.

6. NETI NETI

"Neti-Neti" is a Sanskrit expression that means *"not this, not this"* [3]. It signifies the importance of rejection in identification. Humans are very good in quickly identifying an entity using some distinct part(s); e.g., large stomach and nose in a sculpture allow to reject the hypothesis of that sculpture being of *Natraj*, *Saraswati*, etc., and makes *Ganesh* the most likely answer. This can be thought of as rejection-based identification, where a large hypothesis space is rejected based on the information about some distinct part.

In our case, there are large similarities across different categories as discussed in Sec. 2. This makes it very difficult to perform identification just by learning "what does a sculpture look like?", and thus it becomes very important to learn "what does a sculpture not look like?" as well. In other words, similarity alone can not be a sufficient measure when there is large intra-class variability, and it becomes necessary to perform rejections based on dissimilarity. The problem becomes even more pronounced in the retrieval scenario, where there is no prior knowledge about the class-label associated with each image (i.e., the unsupervised setting), and hence there is no direct way of learning the properties of same/different-class sculptures.

In this work, we have tried to address the above problem by formulating a rejection-based retrieval scheme based on the concept of *Neti-Neti*. We started by rejecting image pixels that are not likely to belong to the carving. Then we formed an image-representation that rejects a large number of patches that are not very discriminative for some given image. Assuming that the most dissimilar images are from different classes, we learnt the metric $\mathbf{w}$ that measures dissimilarity between two images. We then used this metric for extracting pairs of images that are very likely from the same class, and learnt the metric $\mathbf{v}$ that evaluates similarity between two images. During retrieval, given a query image Q, first we ranked all the images based on image-similarity (nearest-neighbour-based retrieval), and then re-ranked only the top few results based on both their similarity as well as dissimilarity (Eq. 5) with Q. In this way, at each step we have tried to emphasize on rejecting the possibililties that might not be useful in the final decision.

7. EXPERIMENTS

7.1 Dataset and Experimental Details

Our dataset contains 524 images from 11 different categories, with $45 - 50$ images per category downloaded from the Internet. We randomly partition the images from each category into around 80% training and 20% query (testing) images. This gives 423 training and 101 query images. The training data is used for metric learning in an unsupervised manner, i.e. without using class information; so the learnt weights $\mathbf{w}$ and $\mathbf{v}$ are least likely to overfit the dataset and

[3] `http://en.wikipedia.org/wiki/Neti_neti`

Method	Descriptor	mAP	P@10
$B1$	SIFT	0.16	0.20
	GIST	0.19	0.27
$B2$	SIFT	0.17	0.21
	GIST	0.26	0.41
$B1$ + Our method	SIFT	0.19	0.25
	GIST	0.24	0.39
$B2$ + Our method	SIFT	0.21	0.28
	GIST	**0.30**	**0.48**

Table 1: Retrieval performance using different methods. See Sec. 7.2 for details.

learning is incremental. From now onwards, we will refer the training images as dataset images (following the practice prevailing in the on-line search scenario).

All the experiments are performed using the split discussed above. To evaluate retrieval performance, we compute the mean Average Precision (mAP) and precision-at-10 (P@10) for each query. These are standard measures for evaluating retrieval performance. The final performance is computed by averaging the scores over all the queries.

7.2 Results

We use the SIFT and GIST descriptors as baseline measures for evaluating retrieval performance. As a first baseline $B1$, given a query image, we compute its descriptor (see the details discussed in Sec. 2), and rank all the dataset images based on the Euclidean distance. As a second baseline $B2$, we manully annotate a rectangular bounding-box around the sculpture in all images (both query as well as dataset images) and then repeat the same process as in $B1$.

To evalute the performance of our method, first we rank the images as in $B1$ or $B2$, and then re-rank only the top 100 images using our approach (as discussed in Sec. 5). While evaluating our method using $B2$ as first stage, we do not perform segmentation as a pre-processing step because the bounding-box itself contains the complete sculpture and performing segmentation over it might help little.

Table 1 shows the mAP and P@10 obtained using different methods. As we have discussed quite a few times, the problem we are addressing is actually a hard problem. This becomes quite evident from the mAP scores achieved by $B1$ and $B2$. Interestingly, GIST performs significantly better than SIFT. This is because using SIFT with discrete BOW to encode information around specific points in an image works best for matching the same instances of an object with mild variations in lighting, view-point, size, etc. Whereas, in our case, all of these are subject to high variations. Also, even with $B2$, the improvement achieved using SIFT is just 1%. In contrary, the GIST descriptor, that encodes a set of perceptual properties, probably better captures the *gist* of a sculpture, and hence performs considerably better than SIFT. With $B2$, the performance for GIST boosts significantly, indicating that it actually provides a better image-representation in our case. These results also validate the observations in Figure 3, which shows that GIST captures the intra-class similarities almost two times better than SIFT.

Our method (i.e., patch-based representation + re-ranking) always improves the performance over the two baselines. Also, performance of our method without using the bounding-box annotations ("$B1$ + Our Method") is better (for SIFT) or comparable (for GIST) as compared to using bounding-box annotations ($B2$). Figure 6 shows some retrieval results obtained by using the best method, i.e. combining our method with $B2$ using the GIST descriptor.

Case	1	2	3
Supervised ML	97.42%	0.85%	0.09%
Unsupervised ML	97.13%	1.56%	0.16%

Table 2: Retrieval performance (%mAP) on toy data using supervised and unsupervised metric-learning (ML) methods. See Sec. 7.3 for details.

7.3 Discussion

To analyze how the unsupervised metric learning approach (presented in Sec. 4.1) compares with the supervised metric learning algorithm LMNN [23], we have conducted a toy experiment that imitates the web-based retrieval scenario (few categories and several distractors). We randomly create 100 samples each from 5 categories (2D Gaussians) with train:test split as 80:20. We consider retrieval on (i) training data, (ii) training data + 10^5 distractors (random samples), and (iii) training data + 10^6 distractors. For the three cases, we get the mAP scores as shown in Table 2. These results support our argument of opting unsupervised retrieval set-up over the supervised one.

Though we have formulated the problem in an unsupervised set-up, it might be interesting to analyze the performance in a supervised scenario. For this, we learn a Mahalanobis metric for the SIFT and GIST descriptors separately using the LMNN [23] algorithm. Using this, we obtain mAP of 0.22 and 0.34 respectively for the two descriptors. Though the results obtained using $B1$ are much less than these, the best results obtained (Table 1, row 5) are very close. This supports the efficacy of our formulation in addressing the challenges involved in the given task.

Also, it is worth noticing that similar to [1], the baseline descriptors do not perform well in our case. However, [1] obtained significant improvements by exploiting the geometry information. Whereas, we do not have the advantage of geometry information and hence the improvements we achieve using our scheme are not appealing, but quite acceptable looking at the complexity of the task.

8. CONCLUSION

In this paper, we proposed a retrieval scheme using a new image-representation and rejection-based re-ranking for the task of unsupervised retrieval of sculpture images given a query image. Using our approach, we were able to achieve better performance than state-of-the art descriptors. Looking at the complexity of the problem, there is a lot of scope for investigating different aspects related to this task. Also, we believe that our method can find applications in various other recognition problems as well.

Acknowledgement

This work was partly supported by Indian Digital Heritage project of DST.

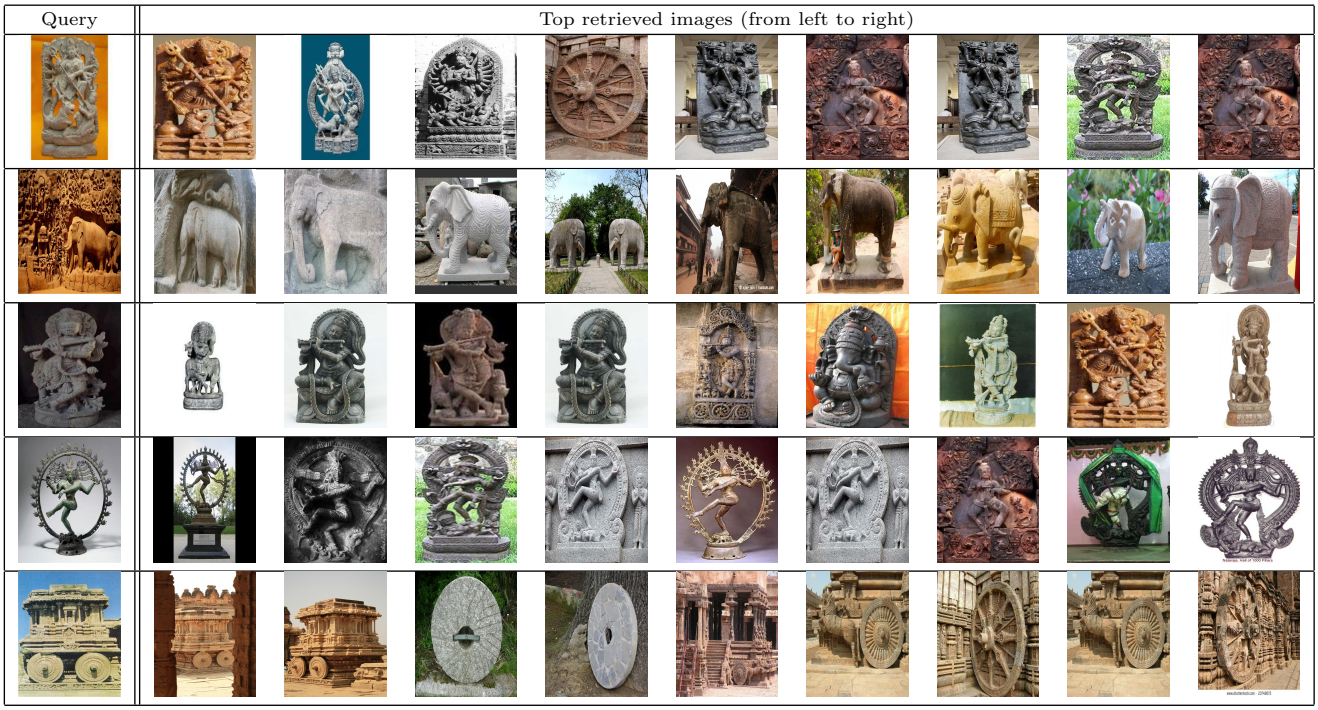


Figure 6: Retrieval results obtained using $B2$ and GIST descriptor combined with our method (see Sec. 7.2 for details). For each query (first column), the top 9 results are shown.

9. REFERENCES

[1] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011.

[2] R. Arandjelović and A. Zisserman. Name that sculpture. In *ICMR*, 2012.

[3] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[4] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *ICCV*, 2011.

[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. In *CVPR*, 2011.

[8] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009.

[9] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2009.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[12] A. Ferencz, E. G. Learned-Miller, and J. Malik. Building a classification cascade for visual identification from one example. In *ICCV*, 2005.

[13] M. Gupta and K. Ramnath. Interactive segmentation tool-box. 2006. `http://www.cs.cmu.edu/~mohitg/segmentation.htm`.

[14] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2007.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.

[18] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011.

[19] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[20] G. Sharma, F. Jurie, and C. Schmid. Discriminative Spatial Saliency for Image Classification. In *CVPR*, 2012.

[21] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, LNCS. Springer, 2006.

[22] A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2010. `http://www.vlfeat.org/`.

[23] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009.