Automatic Ranking of Images on the Web

HangHang Zhang
Electrical Engineering Department
Stanford University
hhzhang@stanford.edu

Zixuan Wang
Electrical Engineering Department
Stanford University
zxwang@stanford.edu

Abstract

We propose a new way to automatically find representative images of a specified object category. Given a large collection of images returned by a web search for an object category, our approach learns the similarities among images without any user help. We extract local features from images, and represent each image using bag-of-words model. The similarity of a pair of image is measured by computing the distance between two signatures. After we get the similarity graph, we run PageRank [3] algorithm to find representative images. We present results and a user evaluation on a variety of object categories, demonstrating the effectiveness of the approach.

1. Introduction

In this project, our goal is to rank images which have the similar semantic meaning. The images can be the results returned by common image search engines or other existing image data sets like ImageNet [4]. Figure 1 shows two examples. The first example shows when we search "stanford gates building" in Google images and the second example shows the first page of vaulting horse in the ImageNet. The current search engines optimize the search results for popular keywords, but the results for less popular keywords are still not organized in a desirable way. We consider the images that share most similarities with rest of the images the most typical image in the collection and rank these images on the top. For example, we do not want the panda image with a large potion of people appears in the first. We would like to see the canonical view of the object on the top. We will mainly develop and test our algorithm using ImageNet, but the algorithm can be applied to other image categories (including Google image, or Flickr). This work is different from the previous work like finding iconic images [2] or re-rank images returned from the search engine [8]. Because the variance of images in each category is very high. Objects in the same category have the same semantic meaning but with quite distinct appearances. It is not practical to use geometric verification to check the consistency between two images, which causes finding the similarity between two images is difficult. To solve this problem, our intuition is the canonical image should share the most features with other images in the same category. We build a graph, in which each vertex represents an image and the weight of the edge reflects the number of common features that two images share. After having the graph, we use PageRank [3] to assign a rank value to each vertex. Vertices with higher rank values are viewed as the canonical images.

(a) Images returned by querying "stanford gates building"

(b) Images of vaulting horse synset

Figure 1. Image collection before ranking

2. Previous work

Recent work in content based image retrieval has mimicked simple text-retrieval systems using the analogy of visual words [19]. Images are scanned for salient regions and a high-dimensional descriptor is computed for each region. These descriptors are then quantized or clustered into a vocabulary of visual words, and each salient region is mapped to the visual word closest to it under this clustering. Each visual word has an inverted index, which records the index of images that contain it. This can improve the speed for the retrieval, reducing the number of images to compare from the size of corpus to the number of images that share the same visual word with the query image. An image is then represented as a bag of visual words, in which tf-idf (term frequency- inverted document frequency) can be used and these are entered into an index for later querying and retrieval. Bag of words model was later applied in natural scene classification [5], content based image retrieval [16] and recognizing landmarks [20]. Experiments show that bag of words model has good performance in these areas.

Fergus *et al*. [6] and Lin *et al*. [11] dealt with the precision problem by re-ranking the images downloaded from an image search. The method in [6] involved visual clustering of the images by extending probabilistic Latent Semantic Analysis (pLSA) [7] over a visual vocabulary. Lin *et al*. [11] re-ranked using the text on the original page from which the image was obtained.

3. Approach

3.1. Outline

We outline our approach for re-ranking images as shown in Figure 2.

1. Collect images corresponds to one specific object.
2. For each image, detect a set of interest regions and extract local feature descriptors.
3. Use PCA (Principal Component Analysis) to reduce the dimension of features.
4. Cluster feature descriptors using hierarchical k-means [14]. Each cluster center represents a visual word.
5. Represent each image by a sparse vector. Compute the distance between pair of images.
6. Compute the PageRank value for each image.

3.2. Bag of words model

We use two types of interest region detectors and in each interest region, we adopt SIFT feature descriptor.

1. *Lowe's DoG detector.* Scale invariant interest regions are detected using Differential of Gaussian detector [12].
2. *Affine invariant detector.* Affine invariant interest regions are detected using Harris-affine and Hessian-affine keypoint detectors [13].

DoG interest region detector is scale invariant. Harris-affine and Hessian-affine interest region detector are affine invariant. For each image, hundreds of local features are extracted. After we get all features from the category, we use PCA to find a subspace to approximate the features so that we can reduce the dimension of features. This step can reduce the computation time for the following clustering. Then we use k-means to cluster those features into vocabulary. Large size of vocabulary tends to have better performance since the visual words are more discriminative when the vocabulary size grows. However, Philbin *et al*. [16] shows that when the size of vocabulary goes beyond a threshold, typically one million, the performance stops increasing. In experiment, we test the performance of our algorithm using different scale of vocabulary.

We use two methods instead of original k-means to generate the vocabulary because when the k is large, each iteration in original k-means takes $O(nk)$ time which is quite slow. The first is to use the hierarchical k-means and the second is to use approximate k-means.

3.2.1 Hierarchical k-means

We use a tree structure to represent the whole vocabulary. If k defining the final number of clusters, b defines the branch factor (number of children of each node) of the tree. First, an initial k-means process is run on the training data, defining b cluster centers. The training data is then partitioned into b groups, where each group consists of the feature descriptors vectors closest to a particular cluster center. We stop this iteration until the height of the tree reaches L. So, we get a vocabulary tree whose size is $b^L = k$. The hierarchical k-means is faster than the original one but it has drawback that when the data point lies close to Voronoi region boundary for each cluster center, the quantization error is large. When the hierarchical structure is built, we can quantize new data point in $O(bL) = O(\log k)$.

3.2.2 Approximate k-means

In original k-means, a large portion computation time is spent on calculating nearest neighbours between the points and cluster centers. We replace this exact computation by an approximate nearest neighbor method, and use a forest of 8 randomized k-d trees [10] built over the cluster centers at the beginning of each iteration to increase speed. The average computation time to find the nearest neighbor in each

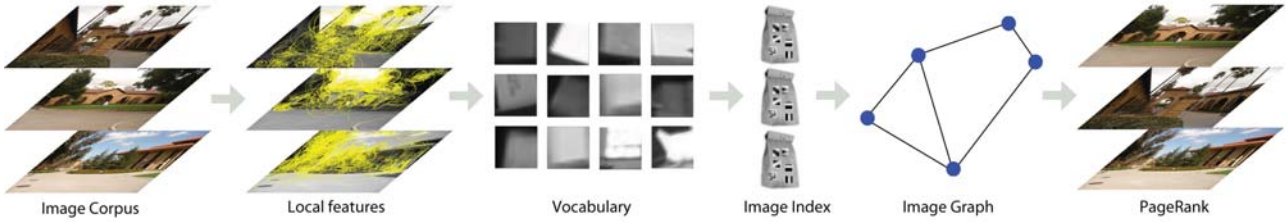


Figure 2. Pipeline of the re-rank algorithm

iteration is reduced from $O(nk)$ to $O(n \log k + k \log k)$, where n is the number of features to be clustered. Usually in a k-d tree, each node splits the dataset using the dimension with the highest variance for all the data points falling into that node and the value to split on is found by taking the median value along that dimension. In the randomized version, the splitting dimension is chosen at random from among a set of the dimensions with highest variance and the split value is randomly chosen using a point close to the median. We combine the output found by these trees to get the best result. A new data point is assigned to the approximately closest cluster center as follows. Initially, each tree is descended to a leaf and the distances to the discriminating boundaries are recorded in a single priority queue for all trees. Then, we iteratively choose the most promising branch from all trees and keep adding unseen nodes into the priority queue. We stop once a fixed number of tree paths have been explored. This way, we can use more trees without significantly increasing the search time. Philbin et al. [16] show that for moderate values of k, the ratio of points assigned to different cluster centers differs from the exact version by less than 1%. After we have k cluster centers, we compute 8 randomized k-d trees for the quantization. The average time for quantizing new data point is $O(\log k)$.

Figure 4 show the parts of visual words we build using hierarchical k-means. To quantize new data point, we use the soft assignment by assigning each feature to its c nearest neighbors. We define the weights to i neighbor as $e^{-\alpha \cdot d_i}$, where d_i is the distance and α is the control parameter. We get a sparse vector representation of the image, in which each entry is the frequency of the particular visual word appears in the image. We normalize the vector to have the sum of 1.

3.3. Similarity of images

After we get the sparse vector representation for each image, we can simply use the inner product of two vectors to estimate the distance between a pair of images. It is fast but it does not consider the relative position of each visual word. Consider the simple example shown in Figure 5, in which each shape represents a visual word. The triangle is closer to the circle than other shapes. Suppose we only have these four visual words: triangle, circle, rectangle and pentagon. Assume image A only contains triangle so its index $I_A = (1, 0, 0, 0)$ and image B only contains circle, $I_B = (0, 1, 0, 0)$ and image C only contains rectangle, $I_C = (0, 0, 1, 0)$. If we compute the inner product, we get the distance between A and B and the distance between A and C are the same. But in fact, the distance between A and B should be smaller due to the triangle and circle are closer to each other.

To take into account of the relative positions of visual words, we compute the EMD (Earth Mover's Distance) [17] between two image signatures. Computing EMD is expensive compare with the inner product, recently Pele *et al*. [15] propose a fast method to compute EMD. Alternatively, we can use pyramid histogram matching [9] to approximate the optimal solution.

The idea of EMD is the following:

In the original EMD paper, a signature is represented by $\{s_j = (\boldsymbol{m_j}, \omega_{\boldsymbol{m_j}})\}$, where $\boldsymbol{m_j}$ is the position of the visual word and $\omega_{\boldsymbol{m_j}}$ is the weight given to this visual word. The integer subscript j ranges from one to the number of features extracted from the specific image. The image index can be viewed as a histogram h_i, in which the vector i index a set of visual words and each visual word has a weight, so the image index is a signature $\{s_j = (\boldsymbol{m_j}, \omega_{\boldsymbol{m_j}})\}$.

Now suppose we have two images $P = \{(\boldsymbol{p_1}, \omega_{\boldsymbol{p_1}}), \ldots, (\boldsymbol{p_m}, \omega_{\boldsymbol{p_m}})\}$ and $Q = \{(\boldsymbol{q_1}, \omega_{\boldsymbol{q_1}}), \ldots, (\boldsymbol{q_n}, \omega_{\boldsymbol{q_n}})\}$. Image P has m features and image Q has n features. $\mathbf{D} = [d_{ij}]$ the ground distance matrix where d_{ij} is the ground distance between visual word $\boldsymbol{p_i}$ and $\boldsymbol{q_j}$. We want to find a flow $\mathbf{F} = [f_{ij}]$ with f_{ij} the flow between $\boldsymbol{p_i}$ and $\boldsymbol{q_j}$ that minimizes the total cost:

$$W(P, Q, \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \tag{1}$$

subject to the following constraint:

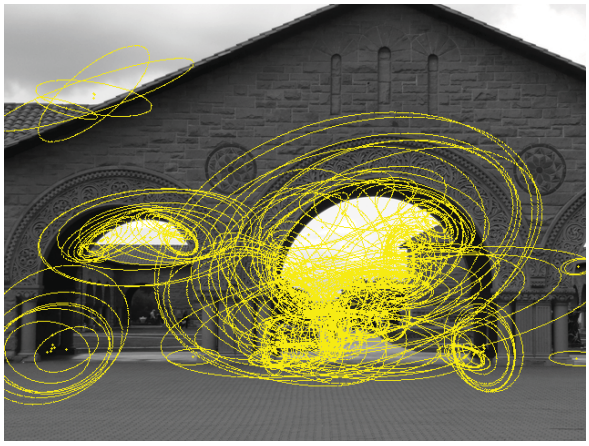
$$f_{ij} \geq 0 \qquad 1 \leq i \leq m, 1 \leq j \leq n \tag{2}$$

$$\sum_{j=1}^{n} f_{ij} \leq \omega_{\boldsymbol{p_i}} \qquad 1 \leq i \leq m \tag{3}$$

$$\sum_{i=1}^{m} f_{ij} \leq \omega_{\boldsymbol{q_j}} \qquad 1 \leq j \leq n \tag{4}$$

(a) DoG keypoint detector

(b) Harris affine keypoint detector



(c) Hessian affine keypoint detector

Figure 3. Different local features

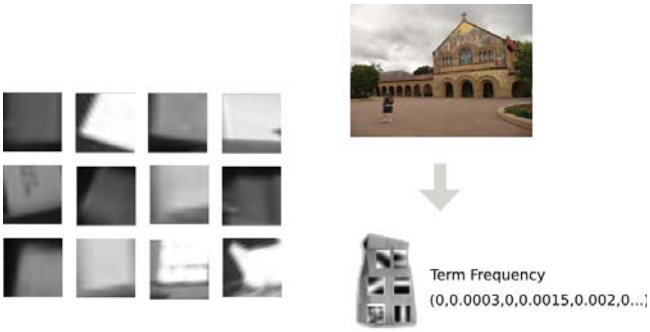


Figure 4. Different local features

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} \omega_{\boldsymbol{p_i}}, \sum_{j=1}^{n} \omega_{\boldsymbol{q_j}}\right) \tag{5}$$

Once the problem is solved, we have found the optimal flow $\mathbf{F}$, the earth mover's distance is defined as:

$$\mathrm{EMD}(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{6}$$

Intuitively, given two signatures, one can be seen as a mass of earth properly spread in space, the other as a collection of holes in that same space. Then, the EMD measures the least amount of work needed to fill the holes with earth. Here, a unit of work corresponds to transporting a unit of earth by a unit of ground distance.

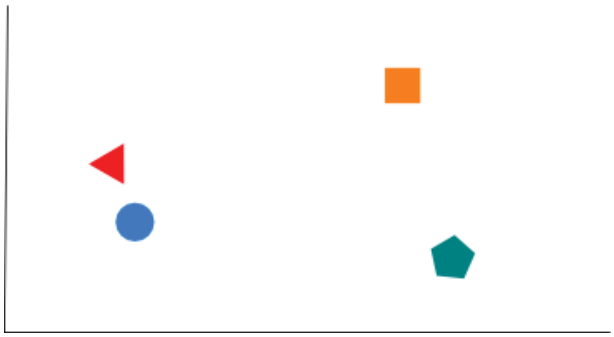


Figure 5. The positions of visual words

3.4. Re-rank images

After we compute the similarity for each pair, we construct a similarity graph. We perform PageRank on this graph and rank the images. We construct the similarity graph as follows: We do content based image retrieval for each image in the corpus and candidate images are returned. We use the methods above to compute the similarity between the query image and the candidate image and add an edge connecting these two vertices in the graph. We define

the similarity matrix $W_{N\times N}$ and $R_{N\times 1}$, where N is the number of images in the corpus. The element in R is defined as the rank value associate with each image. Its stable state is derived from an iterative procedure as follows:

1. Define the similarity matrix $W_{N\times N}$ as:

$$W_{ij} = e^{-dist(i,j)/\sigma^2} \tag{7}$$

where $dist(i,j)$ is the distance metric between image i and image j and σ is the control parameter.

2. Symmetrically normalize W by

$$S = D^{-1/2} W D^{-1/2} \tag{8}$$

where D is a diagonal matrix and $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

3. Do iteration until convergence

$$R(t+1) = \beta S \cdot R(t) + (1-\beta) Y \tag{9}$$

where t denotes the number of iterations and β is the propagating parameter. Y is the initial state representation for each vertex, $Y_i = 1/N$. We set $R(0) = Y$.

We sort images according to their rank values in R.

4. Experiment

4.1. Images of the same object

In the first experiment, we choose ten images from flickr by searching statue of liberty. When we use hierarchical k-means (HKM) and approximate k-means (AKM), we get the same result. The result is shown in Figure 6.

In the second experiment, we take ten photos of the Stanford main quad. The result is shown in Figure 7 and Figure 8.

The running time of two experiments is shown in Table 1.

4.2. Images of similar objects

In the first experiment, we randomly select 30 images from the horn synset from ImageNet. Top 10 images after ranking are shown in Figure 9.

In the second experiment, we randomly select 200 images from the horse synset from ImageNet. Top 20 images after ranking are shown in Figure 10.

In both experiment above, we use inner product to compute the similarity of images since EMD is too slow when the number of visual word is large.

5. Conclusion

5.1. Analyze the algorithm

The current algorithm works on the static landmarks or buildings since the feature we use is suitable for describing the same object. Since we do not segment the object in the preprocessing step, images with complicated background will affect the accuracy of ranking. For the objects that have the same semantic meaning, like pictures in one synset in ImageNet, the SIFT descriptor is not robust enough.

5.2. Future work

To eliminate the affect of background, we can segment the foreground objects first and compute the visual words using the foreground objects. To improve the performance on similar images ranking, we should combine other image features including geometric blur feature [1] or GIST feature [18]. The current EMD algorithm is too slow to rank images in a large scale, we need to integrate the fast EMD algorithm [15] to speed up the procedure of similarity computation. When getting the similarity matrix, we can also adopt graph cut algorithm to cluster images into subcategories, in which images in the same subcategory have strong connection.

References

[1] A. Berg and J. Malik. Geometric blur and template matching. In *IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, volume 1. IEEE Computer Society; 1999, 2001.

[2] T. Berg and A. Berg. Finding Iconic Images. 2009.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[4] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. CVPR*, pages 710–719, 2009.

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from googles image search. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1816–1823. Citeseer, 2005.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM New York, NY, USA, 1999.

[8] Y. Jing and S. Baluja. Pagerank for product image search. 2008.

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

[10] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *IEEE COMPUTER SOCIETY*

Table 1. Running time

	Feature extraction	HKM/AKM	Inner product/EMD	PageRank
Statue of liberty	12.5s	4.3/0.8s	0.1/1.9s	0.1s
Stanford main quad	13.9s	4.7/0.8s	0.1/2.1s	0.1s

Figure 6. Rank result for statue of liberty images using inner product

CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, volume 2, page 775. Citeseer, 2005.

[11] W. Lin, R. Jin, and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *IEEE/WIC International Conference on Web Intelligence, 2003. WI 2003. Proceedings*, pages 242–248, 2003.

[12] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, volume 5. Citeseer, 2006.

[15] O. Pele and M. Werman. Fast and Robust Earth Movers Distances. ICCV, 2009.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, volume 3613, pages 1575–1589. Citeseer, 2007.

[17] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[18] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):300, 2007.

[19] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477. Citeseer, 2003.

[20] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the World: building a web-scale landmark recognition engine. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

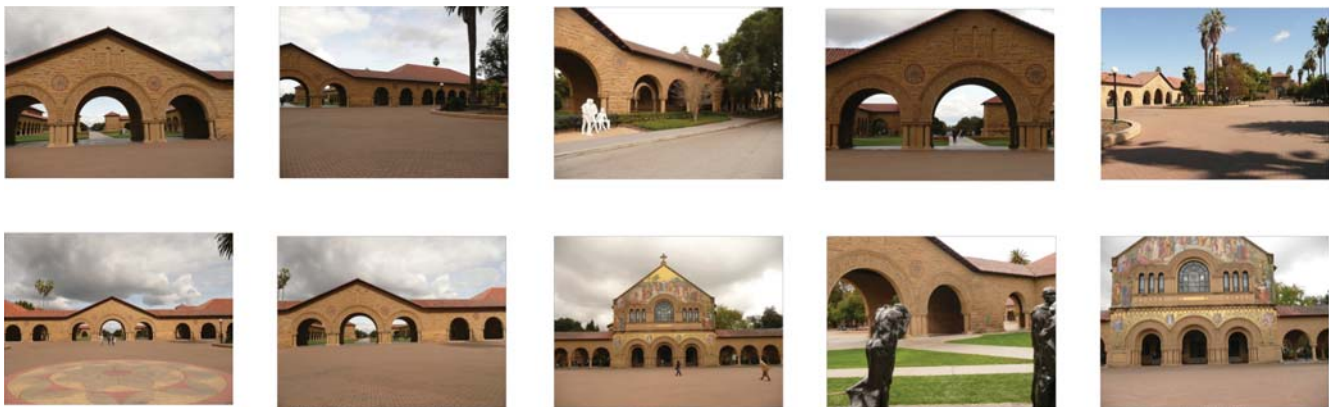


Figure 7. Rank result for Stanford main quad images using inner product

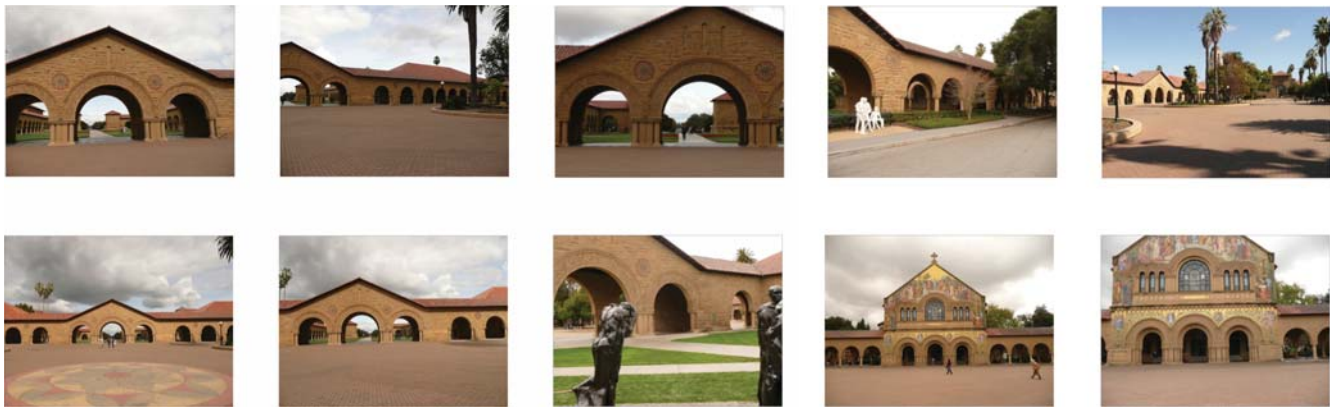


Figure 8. Rank result for Stanford main quad images using EMD

Figure 9. Rank result of horns obtained in ImageNet

Figure 10. Rank result of horses obtained in ImageNet