Combining features to a class-specific model in an instance detection framework

Arnaldo Câmara Lara and Roberto Hirata Jr.
Instituto de Matemática e Estatística
Universidade de São Paulo
São Paulo, Brasil
Email: alara@vision.ime.usp.br
hirata@ime.usp.br

***Abstract*—Object detection is a Computer Vision task that determines if there is an object of some category (class) in an image or video sequence. When the classes are formed by only one specific object, person or place, the task is known as instance detection. Object recognition classifies an object as belonging to a class in a set of known classes. In this work we deal with an instance detection/recognition task. We collected pictures of famous landmarks from the Internet to build the instance classes and test our framework. Some examples of the classes are: monuments, churches, ancient constructions or modern buildings. We tested several approaches to the problem and a new global feature is proposed to be combined to some widely known features like PHOW. A combination of features and classifiers to model the given instances in the training phase was the most successful one.**

***Keywords*-Instance classification; Combining features; Object model;**

I. Introduction

Object detection is a well known studied problem in Computer Vision (CV). It consists in deciding whether there is or there is not an object of some known category in an image or video sequence. A complementary problem is *object recognition* that consists in classifying an object as belonging to a class of a set of known classes. The classifying task usually depends on the construction of a model for each class presented during a training phase [1]. A variation of the problem, called *instance detection*, happens when the elements of a class are formed by only one specific object, person or place. In some sense, all those tasks can be used in real tasks applications as identification of copyright violations, searches in video or image collections, logomark scanning, surveillance or indexing.

In this work, we used the following approach to deal with the instance recognition problem. First, several feature extractors are applied to a training dataset: *Pyramidal Histogram of Visual Words* (PHOW), *Pyramidal Histogram of Oriented Gradients* (PHOG), *Scale Invariant Feature Transform* (SIFT), *Speeded Up Robust Features* (SURF), a new feature extractor called *Pyramidal Patterns of Edges* (PPE) and the *CIE-L*a*b color space histogram*. Some of the extracted features are related to texture, some to the map of edges and the last one is related to the color hue of the objects in the images. Some of the extracted features are then clustered, forming a visual dictionary as in the *bag-of-words* approach [2]. A model is trained using one of the features or a combination of them. The multiclass classifier is solved using the one against all approach [3], an usual strategy when dealing with multiclasses problems. To test and compare the methods, we have collected images from the Internet. There are a lot of databases for visual object detection and recognition [4], [5], [6], [7] but most of them are inappropriate for instance detection and recognition experiments. So, we collected some images under Creative Commons License[1] from Flickr[2] photo-sharing website. The set consists of some famous landmarks around the world: buildings, churches, monuments, palaces, etc. Each class corresponds to only one landmark. To cross-validate the implementation, we also applied the same method in the well-known object recognition database Caltech-101 [6] and we obtained performance similar to the state-of-art methods reported in the literature.

This paper is organized in the following way. In this section, instance detection and recognition problem is presented and we described the contributions of this work. In the next section, the object recognition area and its evolution over time, some works submitted to the 2010 edition of TRECVID and some object recognition databases are reviewed. In Section III, we explain the features used in our work and we introduce a new feature: Pyramidal Patterns of Edges (PPE). Section IV describes the database used to test our framework. Section V explains the experiments, choice of parameters, details of the implementation, results and their analysis. Finally, in Section VI, we state some conclusions and give some directions for future works.

II. Related Work

Object recognition is a well studied problem in CV. A naive Image Processing approach to deal with this task is template matching by pixel correlation. However, it is weak to treat: real scenes, changes in point of views and articulated parts [8]. A better technique uses global descriptors like color and texture histograms [9]. But, again, it is not too robust in real-world scenes with occlusion, cluttered background and

[1] www.creativecommons.org
[2] www.flickr.com

variability of lightning conditions. The latter approach were soon supplanted by part-based methods for their intuitive way to model an object. Those approaches combine local descriptors with their spatial relations. The problem is their computational complexity and also that it fails to learn models of articulated objects. A dataset problem is that ground truths are difficult and costly to build because all parts have to be labeled for training [10].

In more recent years, object recognition researchers started to use a method used previously in texture recognition and text document classification: *bag-of-words* (BoW) model [2]. In short, image local features are clustered, the center of each cluster produces a feature vector that is latter classified to a category word, the whole process forming a dictionary. Image descriptors are formed by frequency histograms of these visual words. No spatial information are kept in the final descriptor: a complete different technique compared to part-based methods. Nowadays, the state of the art in the object recognition includes the bag-of-words approach information about spatial relations of parts of the objects: a return to the part-based models [11], [12].

Instance recognition can be viewed as a special case of object recognition when the members of a class are formed by only one instance of each object. Object recognition is a CV task of great interest to the research community. However, there are not many works focusing on instance detection, categorization or search. The works submitted to the TRECVID 2010's *instance search* (INS) task deal mainly with person instance search and use based on standard object recognition techniques. The work by the National Institute of Informatics (Japan) [13] got the best performance in person queries. It uses a facial descriptor built using face detection and 13 facial points composed by intensity values in a circle of radius 15 pixels. The authors also use local descriptors based on SIFT descriptor [14] and a global RGB histogram. Another team, BUPT-MCPRL [15], uses face recognition methods, HSV color space histogram [16], Gabor wavelet [17], edge histogram [18], HSV color space correlogram [16], local binary pattern (LBP) [19], among other features. A total of 15 teams submitted results in instance search pilot task from TRECVID 2010 competition and the scores for the automatic approach ranged from 0.01 to 0.033 (mean average precision - MAP) [20]. The conclusions stated by the organizers were that the problem is very difficult and the results were very poor.

There are many publicly available databases used for visual object detection and categorization benchmarks in CV researches. They are different in the number of categories, samples per class and purposes. LabelMe database [4] has an interesting tool where users can annotated online the objects in images and videos of the database. The PASCAL Visual Object Classes Challenge [7] is an annual competition with a different database for each year. The 2010 version presented about 11000 images labeled in 20 classes. It consists of difficult images and it is becoming the standard database to test object detection and categorization techniques. The Caltech-101 [6] has 101 image categories with a relative small number of images in each category. The principal use of this database is to evaluate multi-category object recognition algorithms. The Caltech-256 [21] is a natural evolution of the previous version (Caltech-101) with more categories, more variability of the images in each class. The biggest database available today is the ImageNet [5] that uses the hierarchical structure of a lexical database online, the WordNet [22]. It contains more than 12 millions images in almost 18000 categories. SUN Database [23] is a database whose main purpose is to present a large number of scenes categories. It presents almost 900 scene categories (indoor, urban, nature) and about 130000 images. The data used in TRECVID's INS is composed only by videos, is copyright protected and not available publicly.

III. Image Descriptors

In this section, we start reviewing SIFT descriptors. Next, we review PHOW, a technique based on SIFT. We then review PHOG, that is based on oriented histograms. Finally, We introduce a new global feature: pyramidal patterns of edges (PPE). It is implemented using morphological opening operations. Finally, we describe a histogram that uses the CIE-L*a*b color space.

A. SIFT

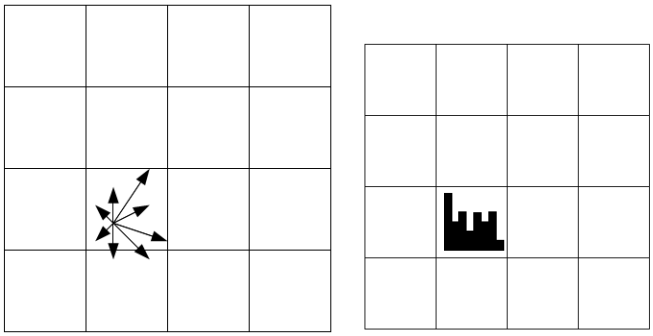
Scale Invariant Feature Transform (SIFT) is one of the most popular feature descriptors used in object recognition problems. Introduced by David Lowe [14], [24], it can be viewed as a keypoint detector and descriptor [1], [19]. Points of interest can be found in different scales of an image. For example, a detail in texture can be visible in a fine scale and the contours of a building are probably visible in a coarse scale. Therefore, points of interest are a function of the position (x, y) and scale s. A convolution by a Gaussian kernel $G(x, y, \sigma)$ is used to compute a specified scale s of an image $I(x, y)$:

$$I_s(x, y, \sigma) = G(x, y, \sigma) * I(x, y). \quad (1)$$

where $*$ is the convolution operator. Usually a kernel with variance equals to $\sigma^2 = s^2 - 0.25$, where s is the scale and 0.25 is an empirical adjustment induced by CCD cameras[25].

The SIFT descriptor has a biological motivation [1]. Biological observations show that neurons in the visual cortex respond higher to some particular gradient orientations. In this sense, SIFT descriptor is a 3D gradient histogram. For each pixel, there are 3 dimensions: its position and gradient orientation. Samples are partitioned in 4×4 square regions and gradient orientations are clustered in a 8-bin histogram for each region totalizing 128 dimensions ($4 \times 4 \times 8$). According to the distance of the center of the square, a weight is applied. Fig. 1 shows an example of a 4×4 square region, the respective 8-bins oriented gradient and its correspondent histogram.

Fig. 2 shows an image of the Opera House in Sydney and some SIFT descriptors plotted in the image with respective orientations, scales and the 4×4 subregions.



(a) 4×4 square regions and 8-bin oriented gradients

(b) spatial histograms

Fig. 1. The computation steps for the SIFT descriptor.

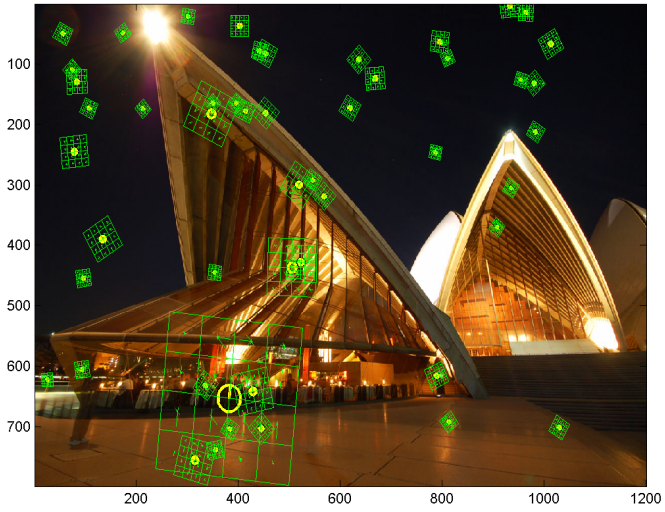


Fig. 2. Image of the Opera House in Sydney with some plotted SIFT descriptors (subregions, orientations).

A variant of the traditional SIFT does not use the keypoint detector. It applies the SIFT descriptor in some random points of the image [26].

B. PHOW

Pyramidal Histogram of Visual Words (PHOW) is an image descriptor based on SIFT descriptors. As introduced in [27], it uses a grid of dense points in the image and for each point of the grid a SIFT is computed. It uses three scales, by default, building a pyramid of descriptors [12]. After this computation, the descriptors are clustered using a k-means algorithm [28] and each cluster is called a *visual word*. The frequency of appearance of each visual word in the dictionary is computed for each image. A frequency histogram of visual words is computed in this step [2].

Finally, the image is divided in increasingly finer spatial subregions and for each subregion the frequency histogram of the last step is computed. The final descriptor is the concatenation of the frequency histograms for all subregions [12] (the entire image is the first level of the division process). This method is in the state-of-art in object recognition [29].

C. SURF

Speeded Up Robust Features (SURF) is a scale and rotation-invariant descriptor proposed in [30]. It consists of a point of interest detector and descriptor. The detector is an approximation of a Hessian Matrix [31] that uses an integral image [32]. The descriptor is based on the response of a Haar wavelet [16] using only 64 dimensions in the final vector descriptor. It improves the computation performance and robustness of the method in relation to SIFT.

D. PHOG

Pyramidal Histogram of Oriented Gradients (PHOG) is a feature descriptor based on *Histogram of Oriented Gradients* (HOG) [18]. HOG was adapted in [33] to be used in different levels, building a pyramid of descriptors. The pyramid is formed dividing the original image in power of two regions for each dimension. So, in level 1, the image is divided into 4 subregions and in level two, into 16 subregions. In each subregion, a histogram of gradient orientations is computed. Generally, 8 bins are used in those histograms. For all levels, the histogram is computed using the finer resolution of the image. The final descriptor is a concatenation of all histograms in each level. PHOG combines shape information through histograms and spatial information through subregions division.

E. PPE

Pyramidal Patterns of Edges (PPE) is an edge-related descriptor proposed in this work. It is obtained by applying a sequence of morphological opening operators [34] in the map of edges of an image using several line structuring elements. A structuring element is a small subset of the domain used to probe the original image and draw conclusions about its shape [34]. The idea is to probe the response of the edges in different angles and lengths.

The first step is to compute the map of edges of the original image. After some experimentation, Canny's method [16] was chosen. The original image is firstly converted to gray-scale and, then, the Canny method is applied.

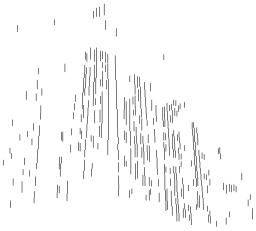
Six different angles (0, 30, 60, 90, 120 and 150 degrees) and nine different sizes (3, 6, 10, 15, 20, 30, 50, 75 and 100 pixels) of structuring elements are used to create 54 line structuring elements (6×9).

An opening operator is applied to the map of edges for each structuring element defined before and the response of each opening is the relative amount of pixels that belongs to the remaining edges. The response of the operator is stronger in the regions that have similar shape to the structuring element used. The process is repeated in three scale levels. In each level, the image size is reduced by a factor of two, so the response of the shapes in the image is measured for each one of the three levels. The final descriptor vector has 162 dimensions(3 levels $\times$ 6 degrees $\times$ 9 sizes).

Fig. 3 shows an image of the Parthenon, the map of edges computed by Canny's algorithm, the opening by a line

(a) Original image.

(b) Map of edges.



(c) Opening result by a line structuring element of size 10 and orientation 90 degrees.

(d) Opening result by a line structuring element of size 100 and orientation 90 degrees.

Fig. 3. Image of the Parthenon, maps of edges and the opening results by a line structuring element of orientation 90 degrees and size 10 and 100, respectively.

structuring element of size 10 and 90 degrees of orientation (response of 0.2496) and the opening by of line structuring element of size 100 and 90 degrees of orientation (response of 0.0051). Only one of the vertical lines in the map of edges has a size greater or equal to 100 pixels.

F. AB Histogram

The color space CIE-L*a*b [35] has a luminosity dimension L and two chromaticity dimensions a and b. The dimension a indicates a color along red-green axis and dimension b indicates a color in blue-yellow axis. In this space, Euclidean distance is adequate to measure the different between two colors. In this method, we first convert an image to CIE-L*a*b color space and then compute a histogram using just two chromacity dimensions. As we are dealing with instances of the same landmark for each class, the color histogram for each image in a class has a great chance to be similar. Differences in color can be due to night or day shots, presence or absence of background objects (people, vehicles, animals, etc). Fig. 4 shows an image of the Cristo Redentor. It was segmented using the dimensions a and b in the CIE-L*a*b color space. As dimensions a and b contain just hue information, we can expected that for two images of the same landmark the AB histogram will be similar in appearance.

IV. Database

Due to the lack of a publicly available image database focusing on visual instance detection and categorization, we collected some images of important landmarks around the World from Flickr web-site. We built an instance dataset where each landmark is a category in the database. All collected images were carefully chosen to have a Creative Commons license. The database has twenty instances and each one has about twenty images. Pictures are viewed from outdoor and they were collected in the highest resolution available. We collected pictures from different views, with different resolutions, at different distances. Some of them were taken at night and some of them have some people. Table I shows the 20 instances. Fig. 5 shows an image for each instance of the database.

(a) Original image of the Cristo Redentor.

(b) Image segmented.

Fig. 4. CIE-L*a*b color space-based segmentation.

TABLE I
Instances of Landmarks database.

#	Landmark	City	Samples
1	Cristo Redentor	Rio de Janeiro, Brazil	21
2	Statue of Liberty	New York, USA	20
3	Eiffel Tower	Paris, France	20
4	Burj Al Arab	Dubai, UAE	21
5	Opera House	Sydney, Australia	20
6	Petronas Towers	Kuala Lumpur, Malasia	21
7	St. Basil's Cathedral	Moscow, Russia	20
8	Congresso Nacional	Brasilia, Brazil	20
9	Big Ben	London, UK	20
10	Capitol	Washington, USA	21
11	Forbidden City	Beijing, China	20
12	Parthenon	Athens, Greece	20
13	Colosseum	Rome, Italy	20
14	Temple of Sagrada Familia	Barcelona, Spain	20
15	Casa Rosada	Buenos Aires, Argentina	20
16	Cologne Cathedral	Cologne, Germany	20
17	Taj Mahal	Agra, India	20
18	Pyramid of Cheops	Cairo, Egypt	20
19	Taipei 101	Taipei, Taiwan	20
20	CN Tower	Toronto, Canada	20

V. Experiments and Results

In this section, we describe implementation details, choice of parameters and experiments. We also analyze the obtained results.

Fig. 5. An example of each instance from the landmarks database. From left to right, top to bottom, they are presented in the same sequence as presented in Table I.

A. Implementation details

All the experiments have been done in Matlab [3] version 7.8 (Release 2009a) 64 bits. We used a Intel Core i5 (64 bits), running on a Windows XP Professional 64 bits Operating System, with 16 GB of RAM memory and 5 TB of disk space.

We used the implementation of SIFT, dense SIFT and PHOW descriptors by VLFeat [36]. VLFeat is a GPL license Computer Vision library that has a lot of modern CV algorithms implemented as SIFT, K-Means, SVM classifier, histogram-related function, among others. It has a Matlab interface and a C API.

We used the implementation of PHOG descriptor available in the site of the authors at http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html. We used the three levels, eight bins and angle equals to 180 degrees as parameters of PHOG descriptor. We choose these values running some tests and using the values reported in literature [27], [33].

AB Histogram was implemented using the Image Processing Toolbox of the Matlab and a fifteen bin parameter for each of the two channels was chosen.

B. Classifiers

We used *Support Vector Machines* (SVM) to train and test the image descriptors. SVM was introduced in [37] and it is one of the most used classifier algorithms in object recognition problems. Given a training set with L training points, each training point being composed by an instance pair (x_i, y_i), $i = 1, ..., L$, where $x_i \in \mathbf{R}^D, \forall i$, and D the number of dimensions of training points, $y_i \in \{-1, +1\}$ [38], [39]. The classification algorithm requires to solve an optimization problem:

$$min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i, \text{such that } y_i(x_i.w + b) - 1 + \xi_i \geq 0 \forall i. \quad (2)$$

If the data were not linearly separable, there is a technique named *kernel trick* that uses special functions, the *Kernel Functions*, that maps the data into a higher dimension space where it can be linearly separable. The correct choice of a Kernel depends on the problem. We tested several kernels and we obtained the best performance using Chi-Squared Kernel that is reported in literature as the best kernel to be used in an object recognition problem [40]. We used the kernel implementation available by VLFeat [41].

[3] www.mathworks.com

TABLE II
ACCURACY OF PPE USING THE DIFFERENT METHOD TO COMPUTE EDGES AND USING RGB CHANNELS OR GRAY-SCALE IMAGE.

Method	Approach	Acc.(%)
Prewitt	RGB	32
Prewitt	Gray	32
Ext.Morph.Gradient	RGB	38
Ext.Morph.Gradient	Gray	30
Canny	RGB	40
Canny	Gray	45

C. PPE implementation

We tested our PPE descriptor in the Landmarks database using three methods to obtain the map of edges: Canny, Prewitt [16] and the external morphological gradient [34] (EMG). Canny and Prewitt return a binary map of edges and EMG returns a gray-scale map of edges. EMG, or $grad_e^{ext}$, is defined by: $grad_e^{ext}(f) = (f \oplus e) - f$, where e is a circle structuring element with radius 1 and $\oplus$ is the dilation operator [34]. The methods were tested using two different approaches:

- *RGB*: the gradient from each of the 3 channels of the RGB color space is computed and the final result is the addition of these 3 intermediate gradients.
- *Gray*: the original image is converted to gray-scale and the gradient is computed.

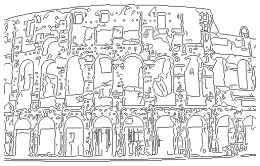
Fig. 6 shows an image of the Colosseum and the maps of edges computed using the different methods. In the example, the original image is firstly converted to gray-scale.

The results of the tests are shown in Table II. Canny's method showed the best performance. We used five images in training set and five in the testing set in this experiment.

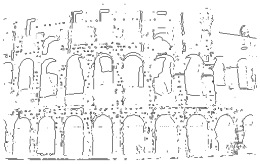
D. Implementation and choice of parameters for PHOW

We used a step parameter with value equal to five for PHOW descriptor. That is the distance in pixels among each sample of the dense SIFT extraction. The region dimension at each SIFT extraction is controlled by a size parameter. We used values of 2, 4, 6 and 10. Therefore, there is an overlap in the

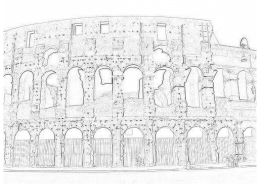
(a) Original image.



(b) Canny.

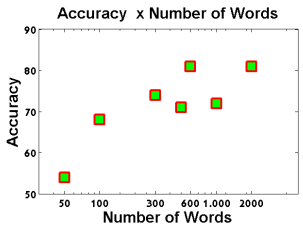


(c) Prewitt.

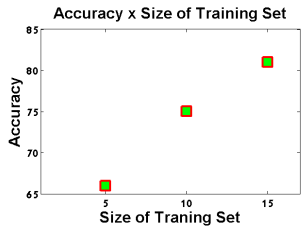


(d) EMG.

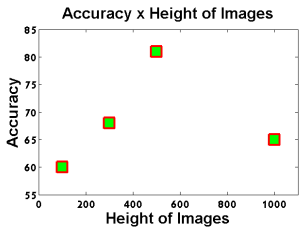
Fig. 6. Three maps of edges of the Colosseum computed using three different methods: Canny, Prewitt and EMG.



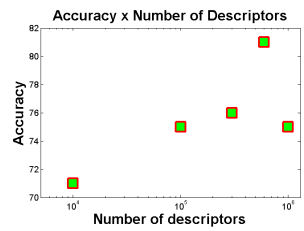
(a) Accuracy x Word size.



(b) Accuracy x Size of Training Set.



(c) Accuracy x Height of Images.



(d) Accuracy x Number of Descriptors.

Fig. 7. It is shown the accuracy x number of words of visual dictionary.

TABLE III
EXPERIMENTS WITH THE CALTECH-101 DATABASE FOR PHOW.

Subset	# Train.	Acc.(%)
Tiny	10	84.00
Tiny	50	96.00
Tiny	100	98.00
Tiny	200	98.80
Tiny	300	98.80
Full	100	69.93

calculation of the dense grid of the SIFT points. We use some empirical tests to choose these values.

We tested two methods to create the visual dictionary in the bag-of-words approach used with PHOW descriptor. In the first method, 100 images are randomly chosen. In second method, 5 images from each of the 20 classes of Landmarks database are randomly chosen. After this, 600000 descriptors are randomly chosen among these images to be clustered to build the dictionary. The second method improves the accuracy in about 8%.

Fig. 7 shows plots of some experiments. We tested some parameters and measured the accuracy in the Landmarks database for PHOW descriptor. The first plot shows the accuracy against the number of words in the visual dictionary. Notice that the accuracy increases until about 600 words, drops around 1000 words and increases again around 2000 words. We chose to use 600 words because it spends less memory. The second plot shows the accuracy versus the number of images in the training set. In all runs the number of images in the test set was 5. Notice that the accuracy increases with the number of images in the training set. The third plot shows the accuracy versus the size of the image. We changed the height of the image keeping the aspect ratio. The best performance was obtained with a height of 500 pixels. The size of an image is directly connected to the size and step parameters of the PHOW descriptor. If size or step parameters were altered, new tests in the size of image must be done. The last plot shows the influence of the number of descriptors used to build the visual dictionary. The best performance was achieved using 600000 descriptors.

To validate our implementation, we run the PHOW descriptor implementation in the Caltech-101 database. We used 2000 words in the dictionary and we tested a subset of the database named “tiny” that uses only 5 classes. A lot of papers report results in this subset [42]. It is a subset with a small number of classes, with easy but hundreds of images. The others classes have just tens of images to be trained/tested. The full database has 102 classes and we rescaled images to 300 pixel in the bigger axis. Results of these experiments are presented in Table III. Notice the increase in the accuracy when it is used more images in the training set. This happens until 200 images in tiny subset.

E. Results of a single descriptor

Table IV shows the accuracy in Landmarks database for each descriptor tested in this work. All runs used 15 images in the training set and 5 images in the test set. Notice that PHOW obtained the best accuracy of all descriptors as reported in literature. The descriptor proposed in this work showed a better performance when compared to the other edge-related descriptors tested in this work: PHOG.

F. Combining descriptors

In this experiment, we combined three types of descriptors: one texture-related, one edge-related and one color-related descriptor. We have chosen the one with the best performance of each type. That is an intuitive way to combine descriptors and to model instances. Some instances are distinguished by their shape, others are distinguished by their texture and other ones by their color. Using a combination of PHOW, PPE and AB-Histogram, we obtained 84% of accuracy in

TABLE IV
ACCURACY OF DESCRIPTORS IN LANDMARKS DATABASE.

Descriptor	Category	Acc.(%)
SIFT interest pts	texture	47
SIFT random pts	texture	41
PHOW	texture	81
SURF	texture	51
PHOG	edges	38
PPE	edge	53
AB Histogram	color	15

TABLE V
ACCURACY OF COMBINED FEATURES CLASSIFICATION USING SVM AND CHI-2 KERNEL.

Class	Accuracy(%)
Cristo Redentor	100
Statue of Liberty	60
Eiffel Tower	80
Burj Al Arab	80
Opera House	60
Petronas Towers	100
St. Basil's Cathedral	60
Congresso Nacional	100
Big Ben	60
Capitol	80
Forbidden City	100
Parthenon	100
Colosseum	100
Temple of Sagrada Familia	80
Casa Rosada	100
Cologne Cathedral	100
Taj Mahal	100
Pyramid of Cheops	100
Taipei 101	20
CN Tower	100
average	84

the Landmarks database. Table V shows the performance for individual class. We used 15 images in the training set and 5 images in the test set. SVM with a chi-2 kernel was used as the classifier. Being the best individual descriptor, the parameters of PHOW were chosen: 600000 descriptors to built the dictionary, 600 words in the dictionary and we changed the size of the images to 500 pixels in height (aspect ratio was kept). Fig. 8 shows the confusion matrix of the classification. The worst performance was in the classification of Taipei 101 that was confounded as Eiffel Tower in 40% of the tested samples.

VI. CONCLUSION AND FUTURE WORK

In this work, we investigated the instance detection and recognition problem. We have reviewed the literature and it says that this is a very hard problem with some additional difficult. One of them is that one can not find an available public image database to test the problem. To mitigate the problem, we collected many images with Creative Commons license from Flick photo-sharing web-site. The images are from landmarks around World as famous churches, tall buildings, modern architecture buildings, ancient monuments, etc. The final dataset has about 20 images for each of the 20 chosen instances of landmarks. We did an exhaustive study testing several image descriptors reported in literature and we

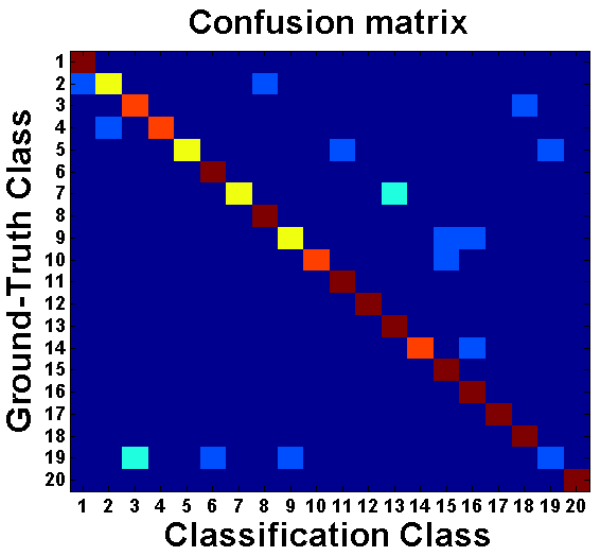


Fig. 8. The confusion matrix of combined descriptors classification (dark blue = 0%, dark red = 100%).

proposed a new one. It uses morphological opening and several line structuring elements to measure the response of a map of edges for specifics angles and sizes of the edges. It showed a performance better than the other edge-related descriptor: PHOG. We combined the best texture-related descriptor, our edge-related descriptor and a color-related descriptor improving the accuracy obtained in the Landmarks database.

Finding alternative ways to combine descriptors and classifiers is a promising area that we are planning to investigate in the future works. We could use weights to measure the importance of each descriptor in the classification of an instance. Another idea is to list the classes that were more confounded and apply some extra classification to improve the performance in these cases. An improvement in the proposed descriptor could be to add spatial information as proposed in [11].

ACKNOWLEDGMENT

The authors are partially supported by CNPq.

REFERENCES

[1] M. Treiber, *An introduction to object recognition: Selected algorithms for a wide variety of applications (Advances in pattern recognition)*. New York, USA: Springer, 2010.

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, 2004.

[3] J. Milgram, M. Cheriet, and R. Sabourin, "One against one or one against all: Which one is better for handwriting recognition with svms," in *10th International Workshop on Frontiers in Handwriting Recognition*. La Baule, France: Suvisoft, october 2006.

[4] C. Russell, A. Torralba, P. Murphy, and T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, Miami, USA, august 2009, pp. 248–255.

[6] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006.

[7] M. Everigham, L. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] R. Nevatia and T. Binford, "Description and recognition of curved objects," *Artificial Intelligence*, no. 8, pp. 77–98, 1977.

[9] C. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "Qbic project: querying images by content, using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases*. San Jose, USA: SPIE, 1993.

[10] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, Rhone-Alpes, France, 2006, pp. 13–38.

[11] P. Felzenszwalb, D. Mcallester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Anchorage, USA, 2008, pp. 1–8.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, New York, USA, june 2006, pp. 2169–2178.

[13] D. Le, S. Poullot, X. Wu, B. Nouvel, and S. Satoh, "National institute of informatics, japan at trecvid 2010," in *Proceedings of TRECVid 2010 - National Institute of Standards and Tchnology*. National Institute of Standards and Tchnology, november 2010.

[14] D. Lowe, "Object recognition from local scale-invariant features," in *The 1999 7th IEEE International Conference on Computer Vision (ICCV'99)*, september 1999, pp. 1150–1157.

[15] X. Guo, Y. Chen, W. Liu, Y. Mao, H. Zhang, K. Zhou, L. Wang, Y. Hua, Z. Zhao, Y. Zhao, and A. Cai, "Bupt-mcprl at trecvid 2010," in *Proceedings of TRECVid 2010 - National Institute of Standards and Tchnology*. National Institute of Standards and Tchnology, november 2010.

[16] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Addison-Wesley Publishing Company, 2002.

[17] L. Costa and R. Cesar Jr., *Shape Analysis and Classification*, 1st ed. CRC Press, 2001.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision and Pattern Recognition (CVPR '05)*, San Diego, USA, june 2005, pp. 886–893.

[19] P. Roth and M. Winter, "Survey of appearance-baed methods for object recognition," Graz University of Technology, Tech. Rep. ICG-TR-01/08, january 2008.

[20] P. Over, G. Awad, J. Fiscus, B. Antonishek, and M. Michel, "Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms, and metrics," in *Proceedings of TRECVid 2010 - National Institute of Standards and Tchnology*. National Institute of Standards and Tchnology, november 2010.

[21] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. CaltechAUTHORS:CNS-TR-2007-001, april 2007.

[22] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, USA: MIT Press, 1998.

[23] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, june 2010, pp. 3484–3492.

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

[25] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[26] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *European Conference on Computer Vision 2006*. Graz, Austria: Springer Berlin / Heidelberg, may 2006.

[27] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, october 2007, pp. 1–8.

[28] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. New Orleans, USA: Society for Industrial and Applied Mathematics, january 2007, pp. 1027–1035.

[29] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. San Francisco, USA: IEEE Computer Society, june 2010, pp. 3360–3367.

[30] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, june 2008.

[31] K. Mikolajczyk and C. Schimid, "Indexing based on scale invariant interest points," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vancouver, Canada: IEEE Computer Society, july 2001, pp. 525–531.

[32] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on In Proceedings of the 2001*. Kauai, USA: IEEE Computer Society, april 2001, pp. 511–518.

[33] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, Amsterdam, The Netherlands, july 2007, pp. 401–408.

[34] E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing*. SPIE Press, 2003.

[35] C. Connolly and T. Fliess, "A study of efficiency and accuracy in the transformation from rgb to cielab color space," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1046–1047, 1997.

[36] A. Vedaldi and B. Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *International Conference on ACM Multimedia 2010*, Firenze, Italy, october 2010, pp. 1469–1472.

[37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, p. 273, 1995.

[38] T. Fletcher, "Support vector machines explained," 2008.

[39] C. Hsu, C. Chang, and J. Lin, *A practical guide to support vector classification*, 2003. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf

[40] Y. Rubner, C. Tomasi, and J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 2000.

[41] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. San Francisco, USA: IEEE Computer Society, june 2010, pp. 3539–3546.

[42] J. Sivic, R. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. San Diego, USA: IEEE Computer Society, june 2005, pp. 370–377.