A mobile visual search application for content based image retrieval in the fashion domain

A. Nodari M. Ghiringhelli A. Zamberletti M. Vanetti S. Albertini I. Gallo
University of Insubria
Dipartimento di Scienze Teoriche e Applicate, Varese, Italy
{*angelo.nodari, marco.vanetti, ignazio.gallo*}*@uninsubria.it*

Abstract

In this study we propose a mobile application which interfaces with a Content-Based Image Retrieval engine for online shopping in the fashion domain. Using this application it is possible to take a picture of a garment to retrieve its most similar products. The proposed method is firstly presented as an application in which the user manually select the name of the subject framed by the camera, before sending the request to the server. In the second part we propose an advanced approach which automatically classifies the object of interest, in this way it is possible to minimize the effort required by the user during the query process. In order to evaluate the performance of the proposed method, we have collected three datasets: the first contains clothing images of products taken from different online shops, whereas for the other datasets we have used images and video frames of clothes taken by Internet users. The results show the feasibility in the use of the proposed mobile application in a real scenario.

1. Introduction

In the last few years the business volume of online shopping is growing up and it also seems that this trend will be maintained in the next future [13]. In particular, the clothing sector has a considerable margin of growth and therefore the number of potential buyers is very high. Nowadays an increasing number of people own mobile phones with Internet access and multimedia capabilities such as built-in camera [1]. Modern search engines offer the state of the art in the text search of content as web pages, databases of documents etc. Regarding querying by images, the scenario is less mature, although the information contained within an image is usually much greater than the information which can be retrieved from the text. The major online search engines offer image search systems even if the greatest practical difficulty lies in sending an image of the desired object. This issue can be overcome through the use of mobile devices able to take pictures of the objects that can be searched through the search services previously cited. A study about the development of applications able to accomplish this task is [21], which shows how a mobile device can exploit the power of queries by image in order to perform generic similarity searches in an ad-hoc database of segmented images.

Figure 1. Global scheme of the proposed method; it is possible to take a picture using a mobile device of a particular dress of interest (a), the query by example is sent to the visual search engine and the retrieved results (b) are returned to the device (c).

There are also some commercial products that make use of images captured from mobile devices in order to perform a visual similarity search, one of the most popular is Google Goggles[1]. It is based on image recognition technology and it is able to identify different classes of objects and returns relevant search results. A major limiting aspect of this software, remarked also by Google itself in the tutorial of the application, consists in the use of this application in the recognition of apparel as it could lead to inappropriate results due to the great generality of this domain. Anyway, one of the main application of this kind of works may be

[1] http://www.google.com/mobile/goggles

found in the online shopping field. The fashion domain is a market sector where it is difficult to exploit text queries in order to search for a particular item. Moreover, the use of an image to search for a particular dress is more descriptive than a text query and sometimes essential because only in this way it is possible to convey some kind of information that cannot be expressed by words or when the user is not even aware of some particular key details necessary to make a suitable query.

Although there are several online visual search services for shopping, there are very few sites that provide the same functionality as mobile services, and they are even more rare in the fashion domain. Amazon is involved in the development of Flow[2], a visual search mobile application derived from the SnapTell[3] tecnology. Picliq[4] by Picitup offers a mobile visual provider for commercial purpose that can be employed to implement specific query by image solutions.

Taking into account the previous considerations the goal is to realize a mobile application able to perform similarity queries by example based on pictures acquired by the camera of a mobile device and in Figure 1 we show a simple use case.

2. Proposed Application

The proposed application is based on a client-server architecture. The client is responsible to present the graphical user interface, where it is possible to choose the dress type and acquire the image, after that it composes the query and sends it to the server.

During the startup phase, an XML file containing the categories, the product names and the focus masks is downloaded using the API provided by the server[5]. The structure of the file is illustrated in Figure 2. The product names are related to a specific product type (shirt, shoes, t-shirt, decolletes, ...), the categories represent a set of product based on the intended use (menswear, womenswear, woman shoes, ...). The XML is used in the category and product name selection interface. The focus masks are images designed to help the user in centering and scaling the object of interest during the image acquisition from the camera of the mobile device. They represent the shape contour of each product name: when the user selects the product name the target cloth belongs to, an adequate mask is displayed and overlapped to the image stream from the camera in order to simplify the acquisition phase and the subsequent image processing, such as object segmentation and the following features extraction. Four examples of focus masks are shown in Figure 3.

[2]http://a9.com
[3]http://www.snaptell.com
[4]http://www.picliq.com
[5]http://www.drezzy.com

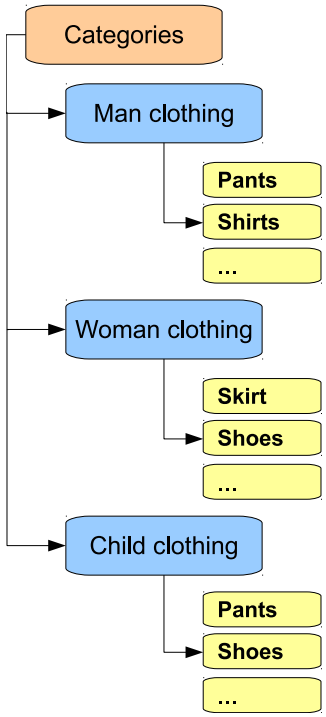

Figure 2. Sample XML schema: The first level elements are the categories and the leaves are the product names.

The acquired image is proportionally scaled in order to reduce the bandwidth usage and the consequently time required to forward the query. The next step consists in the composition of the query, which is encoded in a XML document containing the category, the product name and an ASCII representation of the image obtained through a Base64 encoding[6].

The server is responsible for the object segmentation, the query execution and the composition of the response to send to the client.

First of all, the server application decodes the client request and segments the image taken by the user in order to identify the product of interest into the image. Since the picture is captured using a focus mask, a simple approach is to segment the item cropping the pixels that lie inside the focus mask. The motivation is the need of fast execution time and that solution leads to a sufficient segmentations accuracy with the assumption that the user has used the focus mask to select the object of interest in a correct way. More complex approaches could involve the use of segmentation algorithms like the GrabCut [17] or implementations based on maxflow/mincut like [6].

The next step is the feature extraction for creating a sim-

[6]RFC 4648

Figure 3. Example of the focus masks used to help the user in the selection of the object of interest.

Table 1. Confusion matrix obtained on the *AMF* dataset.

	Hats	Pants	Others	Shoes	Tshirts	PA%
Hats	22	0	0	0	0	100.00
Others	5	132	7	7	7	83.54
Pants	0	0	26	0	5	83.87
Shoes	1	0	0	27	0	96.43
Tshirts	6	1	1	0	22	73.33
UA%	64.71	99.25	76.47	79.41	64.71	

Overall Accuracy (OA): 85.13% **- Kappa-value**: 0.77

ilarity query. In order to search on the database of similar products to the object captured by the user, we extract a set of visual features from the segmented area. The process of the content based image retrieval, delegated to the server, is explained in our previous work [11, 16]. Therefore the server compose the query from the selected category and product name, looking for the products whose visual features are similar to the visual features of the object of interest. All the products are stored in an indexing engine that allows to easily query and rank the results, delegating to the indexing engine the operation of actually computing the similarity measure. All the images in the database are segmented in the startup phase, using a multi neural networks approach [10] [11] in order to extract the visual features only from the area of the image which contains the object of interest.

So, after receiving the request from the client, the server application queries the indexing engine, which returns a list of similar products, then composes them in a XML response and sends it to the client.

The client receives the response from the server and presents the results to the user: an interface with the list of similar products to the captured image, each of them with a thumbnail, its description and its price. If the user wants to see more details he may select the desired item of clothing in order to obtain a more accurate description. From there, it is possible to browse towards the merchant page where it is possible to buy the item.

2.1 Automatic product name selection

Since the manual product name selection may be annoying and somehow counterintuitive, we introduced a classifier trained to automatically select the product name starting from the picture taken by the user. The classifier processes the live stream video from the camera of the mobile device and assigns a specific product name among the classes learned during the training phase. So the user has not to select the product name manually since it is automatically suggested by the system. On the other hand, the category must be chosen once the picture has been captured, since there are a lot of product names shared by different categories, like shoes or pants which can belong to both woman and man apparel, and we found that automatically classify these categories would lead to inaccurate results (and then inaccurate queries which worsens the user experience).

When the user captures a scene, the classifier computes the corresponding class of the framed object of interest. Then the application overlaps the focus mask of the chosen class to the video stream displayed by the device. The effect is that the presented mask changes dynamically as the user frames different scenes with different items. A product name and its focus mask are chosen only if we have a sequence of in agreement classifications longer than a desired threshold, for example 10 frames, in order to avoid instability in the selection of the current displayed mask, due to the possibility that some of the captured frames could be misclassified. This approach avoids also the classification of the stream when the device is moved or shacked by the user before the acquisition of a scene. Another solution to the instability of the displayed focus mask is the addition of an "other" class that acts as a "non-classified" outcome. When a frame is classified as "other", no focus mask is displayed at all.

When finally the user captures a frame, the following process is the same as we have presented in the previous section because the client sends the same request and receive the same response.

3. Experiments

To better understand which algorithm is more suitable to our image classification purpose, we compared some of the best standard algorithms found in literature. We collected the results obtained on the Caltech-101 [15] dataset, which contains 9,146 images belonging to 101 classes. In particular, to have a better overview on this specific problem, we have built a new dataset, called *Drezzy-46*, crawling fashion images from the web, which is composed by 46 classes of different products and for each class there are approximately 100 images, for an amount of about 4600 images in the whole dataset.

Table 2. Comparison in terms of overall accuracy of different images classification algorithms when applied to standard datasets. It is reportd also the average time for a computation of a single image.

	Overall Accuracy %		
Algorithm	Caltech-101	Drezzy-46	Avg. Time
LP-β	82.10	80.12	83.0s
MKL	73.70	88.89	88.0s
VLFeat	65.00	71.30	5.27s
R.Forests*	80.00	-	45.0s
HOG-SVM	28.26	32.90	0.014s
PHOG-SVM	54.00	64.87	0.047s

*the source code is not available

We considered LP-β [12] which is a multiclass classification method, a classifier based on multiple kernel images called MKL [20], VLFeat [19] based on Bag of Words and Random Forests (R.Forests) [3] and an image classification approach based only on R.Forests. We also considered simple approach based on HOG [9] and PHOG [5] features using a SVM classifier [8].

In Table 2 we report a comparison of the above mentioned algorithms on the *Drezzy-46* dataset, we report also the computational time evaluated using a single thread Matlab and C# code, on a Intel®Core™i5 CPU at 2.30GHz.

Starting from this initial analysis and from the considerations found in literature, our goal is to simplify the existing models in order to obtain a more performant algorithm suitable to be executed on a mobile device. For this reason we decided to not use LP-β and MKL since they require, on our platform, days for the training phase, minutes to classify each image during the testing phase and gigabytes of disk space to store all the features data extracted from the training dataset. The R.Forests algorithm is expensive in term of computational time since it requires to compute both PHOG and PHOW [4] features descriptors for each image.

A deep analysis of the features PHOG, PHOW, VSSIM [18] and G.BLUR [2], used in the presented algorithms of Table 2, shows how the use of a single SVM with a PHOG feature represents the best compromise between speed, performance and complexity. In particular we chose a regularized SVM which uses radial basis functions kernel, with the configuration parameters $C = 1$ and $\gamma = 0.01$. The other parameters were automatically selected through a K-fold cross validation system with $K = 5$.

The Content-Based retrieval and indexing engine and the related server application has been already evaluated in our previous work [11] so we analyzed only the automatic product name classification phase designed through the API for the client application.

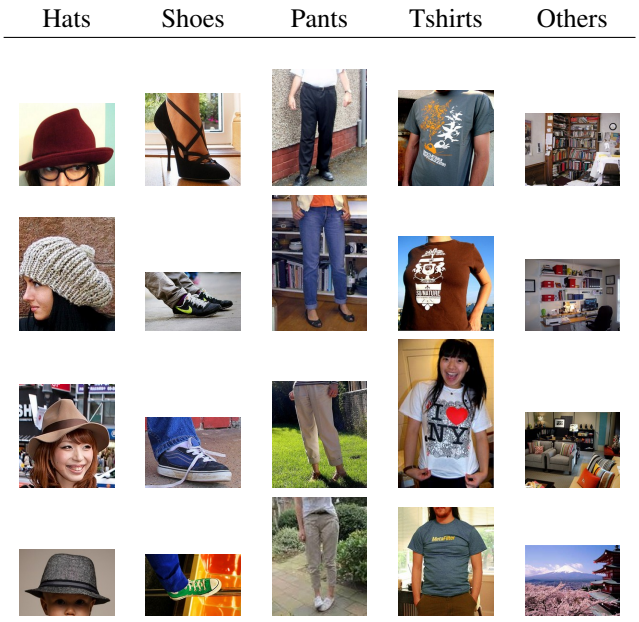

Figure 4. Three image examples from each class of the *AMF* dataset used for the classification task.

In order to test the performance of the classification approach, an ad hoc dataset *Artelab Mobile Fashion (AMF)* was created, some examples are showed in Figure 3, with 4 object classes *Hats, Shirts, Pants, Shoes* and a fifth class *Other* with only the background or objects which do not belong to one of the previous considered classes. For each object class 100 images were collected, and 400 images for the class *Other*, 800 photos in total, mainly crawled from internet. We chose real-world images, with heterogeneous light conditions, quality of the image, different capturing devices and various represented subjects. All the images were resized proportionally fixing the larger side to 200 pixels. The dataset collected was uploaded to our homepage [7] in order to be used by future works for comparison.

The PHOG parameters were tuned in order to find the best compromise between computational performance and classification accuracy, as showed in Figure 6. In particular, 13 bins for each histogram and 3 levels of the pyramid were chosen. Higher values for these two parameters did not produce significant benefits in accuracy but exponentially increased the computation time.

We use Overall Accuracy (OA) and Kappa [7] as metrics for the evaluation phase and the confusion matrix in Table 1 shows the result obtained on the test set.

Taking into account the classification results of the pre-

[7] http://artelab.dicom.uninsubria.it

Table 3. Number of frames for each test video used. Each video contains frames belonging to the 5 classes of interest.

	Hats	Tshirts	Pants	Shoes	Others
Video0	23	37	40	44	53
Video1	42	47	36	29	27
Video2	51	33	39	46	44
Video3	47	25	29	61	72
Video4	0	32	47	27	63
Video5	35	47	41	51	0
Video6	34	37	0	0	115
Video7	0	40	34	41	106
Video8	31	46	37	22	27
Video9	36	31	49	24	65
TOTAL	299	375	352	345	572

vious experiment, we have subsequently evaluated the proposed method on a video stream sequence. By using a smartphone we acquired 10 short videos lasting between 16 and 23 seconds building a new dataset called *Artelab Video Mobile Fashion (AVMF)*. These videos are recorded in different environments such as home and office rooms, and show clothes belonging to the four classes presented in the previous experiments, worn by different people.

From each video, 10 frames per second were extrapolated, for an average of 194 images from each video, proportionally scaled to 200 pixels and then divided in each of the 5 classes considered in this study to obtain the truth used in the evaluation process. The distribution of total 1943 frames obtained for each video is shown in Table 3.

The classification results in terms of Overall Accuracy and Kappa obtained from the 10 videos are shown in Table 4. The average accuracy among all the videos is 74.85%. We captured the videos trying to vary the clothing that shall be recognized and the background context, consciously creating, in some cases, details that have affected the classification accuracy. We have always maintained a good video quality and good lighting.

The entire process of extraction of the PHOG feature from the captured image and prediction of the class with the trained SVM needs, on the average, 260ms on a ZTE Blade with 600MHz single-core ARM cpu and 150ms on a Samsung Galaxy SII (GT-I9100) with 1.2GHz dual-core ARM cpu. These results show that the proposed apporach is able to guarantee a real time service on an average mobile device [14].

3.1. Conclusions

In this study, we have presented a mobile application that allows to perform visual search queries in the fashion domain using a client server architecture. Starting from a picture, the application automatically classify the object of interest in order to facilitate the user to compose a query, minimizing his interaction with the device.

The combination of a SVM and a PHOG feature for the classification task, shows high discriminative capacity and its low computational cost makes it a perfect candidate for the use on devices with low computational power and in a real time context. In particular, we process the video stream from the built-in camera of the mobile device, in order to suggest an automatic prediction of the product name of the user's object of interest. The query result consists of a set of commercial products, related to the object of interest, returned using the API of an online visual indexing engine.

Moreover we are working to extend the application in order to recognize other types of clothing and at the same time increasing the performance of classification.

Table 4. Classification accuracy of extracted frames from the *AVMF* dataset. We report the Overall Accuracy and the Kappa value for each extracted video sequence.

	Overall Accuracy (%)	Kappa
Video0	77.66	71.00
Video1	73.18	68.00
Video2	76.47	70.00
Video3	71.86	65.00
Video4	74.31	67.00
Video5	76.93	69.00
Video6	72.89	64.00
Video7	77.38	71.00
Video8	72.21	66.00
Video9	75.61	65.00
Average	74.85	68.00

Figure 5. Example of video frames from the *AVMF* dataset used in this study for each category take into consideration in this study.

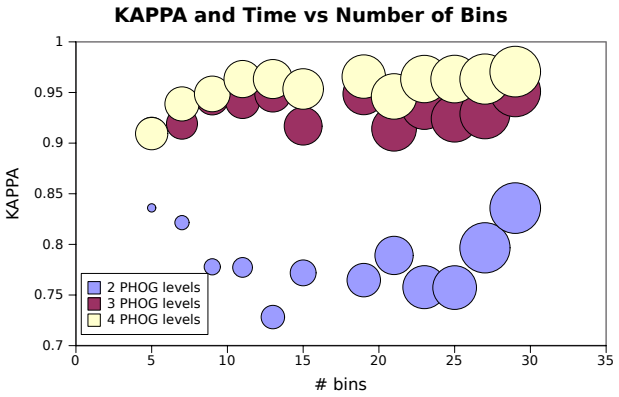

Figure 6. The plot shows how the execution time (diameter of the circles) and the Kappa value depend on the two parameters: *numebr of bins* of the PHOG histogram and *number of layers* of the same feature.)

References

[1] Yearbook of statistics chronological time series 2001-2010, 2011. Telecommunication Development Bureau.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *In CVPR*, pages 26–33, 2005.

[3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *IEEE International Conference on Computer Vision*, 2007.

[4] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007.

[5] A. Bosch, A. Zisserman, and X. Muoz. Representing shape with a spatial pyramid kernel. In *Conference on Image and Video Retrieval*, 2007.

[6] J. Carreira and C. Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010. description of our winning PASCAL VOC 2009 and 2010 segmentation entry.

[7] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[8] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages 886–893, 2005.

[10] I. Gallo and A. Nodari. Learning object detection using multiple neural netwoks. In *VISAPP 2011*. INSTICC Press, 2011.

[11] I. Gallo, A. Nodari, and M. Vanetti. Object segmentation using multiple neural networks for commercial offers visual search. In *EANN/AIAI (1)*, pages 209–218, 2011.

[12] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228. IEEE, 2009.

[13] L. Khakimdjanova and J. Park. Online visual merchandising practice of apparel e-merchants. *Journal of Retailing and Consumer Services*, 12:307–318, 2005.

[14] S. M. Kuo, B. H. Lee, and W. Tian. Real-time digital signal processing implementations and applications. 2006.

[15] R. P. L. Fei-Fei; Fergus. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 28:594–611, April 2006.

[16] A. Nodari and I. Gallo. Image indexing using a color similarity metric based on the human visual system. In *International Conference on Machine Vision, Image Processing, and Pattern Analysis (ICMVIPPA 2011)*, 2011.

[17] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. volume 23, pages 309–314, 2004.

[18] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007.

[19] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In A. D. Bimbo, S.-F. Chang, and A. W. M. Smeulders, editors, *ACM Multimedia*, pages 1469–1472. ACM, 2010.

[20] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[21] T. Yeh, K. Grauman, K. Tollmar, and T. Darrell. A picture is worth a thousand keywords: imagebased object search on a mobile platform. In *Proceedings of CHI 2005, Zhao, R, Grosky, W. 2002, Bridging the semantic gap in image retrieval, Distributed Mulitmedia Databases: Techniques and Applications, T.K. Shih (ED.), Idea Group Publishing*, 2005.