Object Segmentation using Multiple Neural Networks for Commercial Offers Visual Search

I. Gallo, A. Nodari, and M. Vanetti

University of Insubria, Dipartimento di Informatica e Comunicazione
via Mazzini 5, 21100 Varese, Italy
{ignazio.gallo,angelo.nodari,marco.vanetti}@uninsubria.it

Abstract. We describe a web application that takes advantage of new computer vision techniques to allow the user to make searches based on visual similarity of color and texture related to the object of interest. We use a supervised neural network strategy to segment different classes of objects. A strength of this solution is the high speed in generalization of the trained neural networks, in order to obtain an object segmentation in real time. Information about the segmented object, such as color and texture, are extracted and indexed as text descriptions. Our case study is the online commercial offers domain where each offer is composed by text and images. Many successful experiments were done on real datasets in the fashion field.

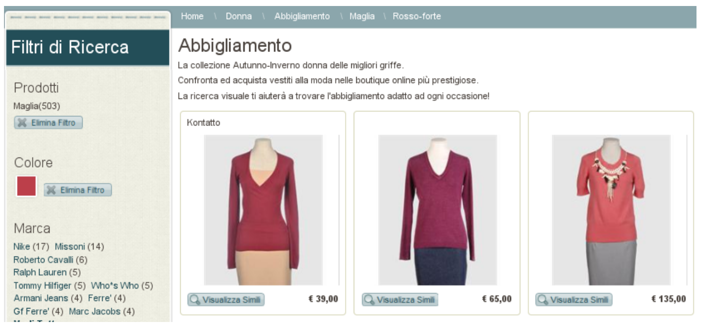
Keywords: visual object segmentation, visual search, multiple neural networks

1 Introduction

In e-commerce applications, information relating to a product are found in the textual description but in many cases it is often difficult to express in words what you want to buy, then it is much easier to choose an object watching the images associated to each commercial offer. In [1, 2] many different region-based and semantic CBIR systems, that allow search with sample images, are mentioned. However, a typical search on an e-commerce site should not start with a query based on a sample image, so it is necessary to provide the user with new navigation tools, such as multi-dimensional navigation. Our method extends the common CBIR tecniques using an object class image segmentation strategy in order to combine multi-dimensional navigation and navigation by examples. Object class image segmentation aims to assign predefined class labels to every pixel in an image and therefore is a crucial step in order to find the Object of Interest (OI) in the image and then its basic features required for multidimensional navigation, like color and texture.

Recently, some applications are emerging, specialized in visual search technologies that lets people hunt online for bargains using pictures of clothing, handbags, shoes or other items they might desire. For example Google's acquisition of Like.com, was seen by some as a competitive response to Bing, the Microsoft search engine touted as a “decision engine” for shoppers. These applications are built on sophisticated machine learning and visual recognition technology and for business reasons all the details of their implementations are not revealed, so it is very difficult to make a comparison. As a demonstration of this, in [3] Skopal analyzed all the systems listed at Wikipedia[1], and declares that in many cases information about the technology used in these engines is not available.

In this work we propose a visual search web application, that takes advantage of new computer vision techniques and is deployed in a commercial version named Drezzy[2]. Figure 1 reports a screenshot of the web interface.



Abbigliamento

La collezione Autunno-Inverno donna delle migliori griffe.

Confronta ed acquista vestiti alla moda nelle boutique online più prestigiose.

La ricerca visuale ti aiuterà a trovare l'abbigliamento adatto ad ogni occasione!

Fig. 1: A screenshot of the application made using the method proposed in this article. On the left column an example of color entities identified by the use of the proposed approach. The images shown on the right are the search result obtained by selecting “red” as color and “sweater” as product.

2 The Proposed Application

The basic idea of our application is the segmentation of OI in images associated with commercial offers, using a new technique based on a set of neural networks (see section 2.1). From each segmented object we can extract several information and in this paper we analyze color and texture (see sections 2.2 and 2.3). These visual information is then transformed into a linguistic description. The main advantage of this representation is the possibility in the direct use of indexing and retrieval tools for text mining. Moreover other visual information can be extracted, such as the shape of the object or its subparts.

For the color and texture extraction we used a Machine Learning method, in particular a Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel, as it provides state of the art for many classification problems [4].

[1] Wikipedia: List of CBIR engines [Online; accessed 2-April-2011]

[2] `www.drezzy.com`

2.1 Object Segmentation

In this section we present the object segmentation model used in this application. The model, called Multi-Networks for Object Detection (MNOD), has already presented [5] as a tool for objects detection and is here adapted to the more complex objects segmentation problem. We chose to use a model based on supervised neural networks because neural models have a high generalization ability and good robustness, in this way the same type of performances are always guaranteed varying the classes of objects to be segmented.

The MNOD model consists of a tree of neural networks, where each network uses a sliding window of size W_S to read the features extracted from the input image. Each MNOD node $C^n_{I_S,W_S}(F_1,\ldots,F_m)$ extracts an input pattern for the neural network by concatenating the contents of the sliding window, placed over each resize input image $(F_1,\ldots,F_m)$, according to a second parameter I_S. A real example of configuration created for our application is shown in Figure 6, while in [5] you can find many details on the segmentation process that we have here omitted for lack of space. A node produces in output a soft classification map (segmentation map) where the pixels containing higher values identify the objects of interest. Only the root node of a configuration is interpreted as the segmentation map for a given object class, using a threshold in order to obtain a binary map. In the present work, the structure of a single node consists in a feed-forward Multi-Layer Perceptron (MLP), trained using the Rprop (Resilient Backpropagation) learning algorithm proposed by Riedmiller and Braun [6].

The particular aspect of this model lies in the connection between nodes, which in practice means that the output of a node becomes the input of a parent node. This means that some features of $(F_1,\ldots,F_m)$ could be the output of child nodes. Regarding the tree structure, links between nodes at different levels serves only to indicate which nodes make use of the output of a particular node.

The features set, such as information on edges, color, etc., can be directly extracted from the input images. In this work we used the normalized red ($RN_s = R/(R+G+B)$), green ($GN_s = G/(R+G+B)$) and blue ($BN_s = B/(R+G+B)$), and the Brightness (Br_s) as color features. Information about edges were extracted using a 1-D vertical (EV_s) and horizontal (EH_s) derivative filter (using the mask $[-1,0,1]$) and a simplified version of the *Histogram of Oriented Gradient* (HOG) [7] feature descriptors, here Oriented Gradient ($OG_{b,s}$). We first compute the gradient values and then, instead of creating the cell histograms as in [7], we create b input maps dividing the gradient orientations in b intervals. The s parameter, used in all the features, represents the scale at which the input image is resampled, before the feature computation. Figure 2 shows an example of all the features used in this work. Figure 6 shows a real configuration with 4 levels, the figure also shows the segmentation map of the OI identified.

The search space for an optimal configuration of the MNOD is very large because we have to choose the depth of tree, the number of nodes for each level, the best parameters and the features most suitable for each node. To restrict the search space, and make the configuration process much faster, we fix the depth to

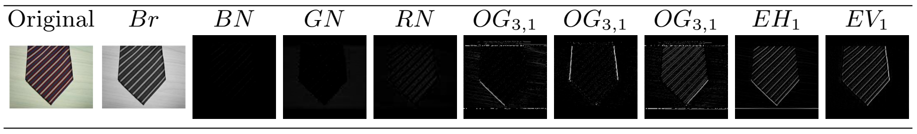


Fig. 2: An example of features extracted for MNOD. The input image is showed on the left. In our experiments, each node can be configured with a subset of this features.

four because we have noticed experimentally that adding more than four layers, the accuracy improves very slightly. In addition, we fix the number of nodes to 4 in the first level and 3 in the two intermediate levels. We have also observed experimentally that using small values of W_S and large I_S values for all the nodes in a layer, alternated by large W_S and small I_S in the subsequent layer, results in a better segmentation accuracy. For this reason, we configure each segmentation model by choosing the parameters for each level between the following subsets $I_S = \{50, 90\}, \{10, 30\}, \{40, 70\}, \{30, 70\}$ and $W_S = \{1, 3\}, \{7, 9\}, \{1, 3\}, \{3, 5\}$, where the first subsets of parameters on the left are for the first level and the last on the right for the output level. To speed up the configuration process we used only nodes pre-configured with the following subsets of features: $C^n_{I_S,W_S}(OG_{b,s})$ choosing $b \in \{3, 5\}$ and $s \in \{1, 0.5, 0.25\}$, or $C^n_{I_S,W_S}(EV_s, EH_s)$ choosing $s \in \{1, 0.5, 0.25\}$, or $C^n_{I_S,W_S}(OG_{b,s},\ RN_s,\ GN_s,\ BN_s)$ choosing $b \in \{3, 5\}$ and $s \in \{1, 0.25\}$, or $C^n_{I_S,W_S}(OG_{b,s}, Br_s)$ choosing $b \in \{3, 5\}$ and $s \in \{1, 0.5, 0.25\}$.

2.2 Object Colors

Our method, to extract the representative colors from the images, is based on a previous work [8] in which we explained an efficient and innovative approach to represent the information about colors using a textual representation. We have selected 27 representative colors based on the natural language color descriptors derived from the ISCC/NBS system as proposed by the Inter-Society Council[3]. The set of selected colors has been created in order to obtain a significative discretization of the entire RGB color space in this application domain. This set of colors is used to extract the principal colors and their quantity in the image considering all the pixels that belong to the OI identified by the segmentation mask.

The first step consist in an image quantization, associating every RGB pixel of the OI with one of the 27 principal colors. In such a way the result is an histogram of representative colors. As shown in the Figure 3 most of the pixels belonging to the OI are assigned to the color "light gray".

The second step consists in the transformation of the principal colors, taken from the previous step, in a textual description suitable for an efficient indexing. In order to keep the information on the amount of color extracted each color

[3] `http://www.iscc.org/`

name is repeated in according to its occurrence in the image in such way it is possibile to use, in retrieval, the term frequency as a measurement of color quantity. One of the major problems encountered in color extraction from images arises from the metric adopted to evaluate the distances between colors. The Euclidean distance on the RGB space, or one of its linear transformations, is inadequate to represent the human visual similarity, so we have adopted an approach based on a similarity metric estimated on human visual perception. We have collected a set of 2500 user judgements, every of them represents an association between the RGB space and one of the 27 principal colors selected. Using the collected data, we have trained an SVM which takes in input an RGB value and has in output one of the 27 principal colors. In this way a prediction of the trained SVM corresponds in a quantization of the RGB space in the subset of colors previous explained.

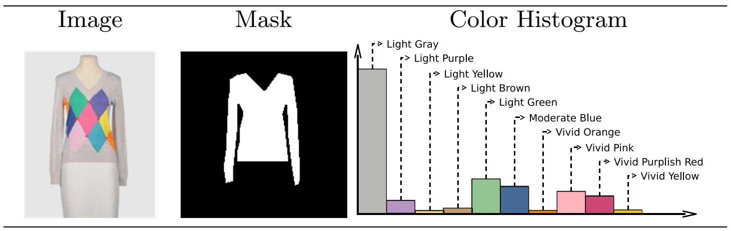


Fig. 3: An example of color extracted from an offer's image in according to the set of pixel belonging to the OI showed in the mask image. The histogram represents the amount of colors contained in the OI. For each histogram's bin the textual descriptor of the representative colors is displayed.

2.3 Object Textures

The texture extraction task is designed as a classification problem. Dealing with a massive amounts of offers, a key requirement of the system is the extraction speed for visual features. The texture classes with wich we deal presents strongly spatial oriented patterns, this is why an extremely simple texture descriptor, based on convolution filters, is powerful enough to separate the problem classes.

Processing the original image with a convolution filters bank (see Figure 4), we obtain the maps $M_1, M_2, M_3, M_4, M_5, M_6$ from which the final features are extracted as follows: $f_k^{max} = \sum_{p \in I} \max(0, M_k(p))/|I|$,
$f_k^{min} = \sum_{p \in I} \min(0, M_k(p))/|I|$, with $k \in \{1, 2, 3, 4, 5\}$ and
$f_6 = \sum_{p \in I} \max(0, M_1(p), M_2(p), M_3(p), M_4(p))/|I|$, where I represents the set of points contained in the OI segmentation mask.

The first four maps are obtained using standard edge detection filters (convolution kernels are shown in Figure 4), M_5 is obtained using a custom convolution kernel and M_6 is computed by taking the maximum punctual value from the first 4 maps.

The feature set is small (a total of 11 features), but in our domain contains important visual information able to discriminate the chosen texture classes. An SVM classifier was trained using the *color-texture* dataset and using a 5-fold auto-configuration strategy to tuning the SVM parameters.

The result of the texture extraction process is a linguistic label that corresponds to the class predicted by the classifier. The label thus obtained will be inserted into the indexing engine.

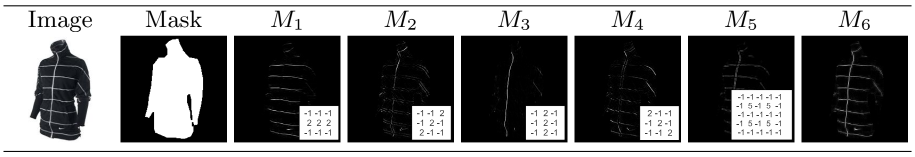


Fig. 4: An example of the input image, the segmentation mask containing the OI and the results of the convolution filters used to extract the texture features, together with corresponding convolution kernels.

3 Experiments

The application we propose has been evaluated on different datasets in order to isolate and measure the performance of the three main phases. In a first experiment (see section 3.2) we evaluated the neural segmentation algorithm that underlies the other two following steps. In particular in the section 3.3 we evaluate the automatic extraction of color and in section 3.4 we evaluate the automatic extraction of the main textures. A detailed comparison with other similar segmentation methods available in literature, was done in [5]. The two phases that follow the objects segmentation have been evaluated independently, extracting the information of color and texture only from the pixels belonging to the OI. That means in practice to involve a mask that identifies the object, created by hand for each image in the dataset.

All the tests reported in the following sections were performed using a single thread C# code, on an Intel®Core^{TM}2 Duo processor T8100 @2.10Ghz.

3.1 Datasets

Analyzing the literature we realized that there are no image datasets available to evaluate an application that deals with objects segmentation in the fashion field. For this reason we create seven datasets of images, each of which contains images of one of the following categories: *Underwear*, *Lingerie*, *Menswear*, *Womenswear*, *Tie*, *Shoe* and *Hat*. Each dataset contains about 200 images, divided into train and test subsets. Figure 7 contains some representative examples of each dataset.

To evaluate the performances in color and texture retrieval we have built the *color-texture* dataset containing nine classes of texture with a wide range of colors, typical in the domain of women's clothing. The texture classes represent the typical decorations found on tissues, in particular we chose horizontal lines (*horiz*), *leopard*, polka dots (*polka*), *rhombus*, *squares*, vertical lines (*vertic*), *zebra* as texture classes. The solid color (*color*) and "*other*" classes were also been included to represent, respectively, tissues with homogeneous color and with decorative patterns that do not fall into the 7 textures listed above. The images in these datasets were also manually labeled associating at each image the most representative colors. In Figure 5 we show some sample images highlighting the manually segmented OI.

All the images in the datasets used in this work, were taken from the Drezzy application and are available for research purposes[4].

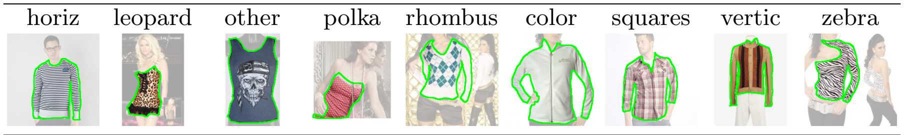


Fig. 5: Image examples taken from the *color-texture* dataset. The OI class was segmented by hand.

3.2 Object Segmentation

In this experiment we evaluated the performance of our segmentation algorithm by applying it to the seven datasets presented in Section 3.1. In order to evaluate segmentation methods we adopted the VOC2010 [9] perclass measure.

For each of the seven classes we trained an MNOD model using the configuration process described in Section 2.1. An example of MNOD configured for the dataset "Menswear" is shown in Figure 6, which also shows the maps produced by each node for a given input image, and parameters and features with which each node has been configured. Finally, we applied the trained models to the images of each test set and the results are shown in Table 1.

We have obtained high quality segmentation results using the trained MNOD model proposed and in Figure 7 we have included some examples for each image data set. The use of these segmentations facilitates the extraction of visual features such as color, texture, shape, etc.. These results lead to significant advantages in terms of quality of the extracted features, which are related only to the OI and are not influenced by the background. The configuration with

[4] `http://www.dicom.uninsubria.it/arteLab/dataset.html`

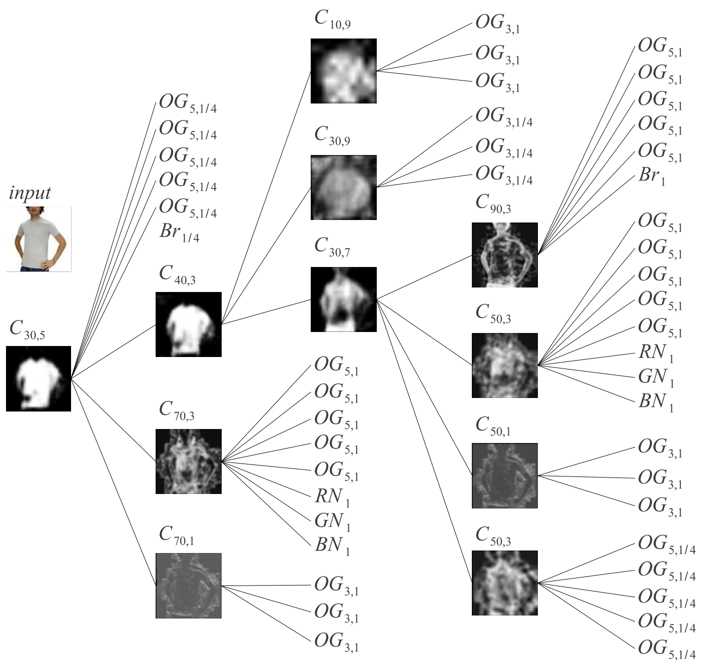


Fig. 6: MNOD configuration used for the dataset "Menswear". The maps produced by each node $C^n_{I_S,W_S}(F_1,\ldots,F_m)$ and the features used in input, all related to the same test image, are showed.

the set of constraints established is nearly automatic, and then requires little time. Moreover, numerical results shown in Table 1 show that our algorithm can always be reliable regardless the class of objects on which you choose to work.

The computational complexity of the segmentation algorithm was computed on the seven data sets we used in our experiments. The average time to process a never seen image is $1196ms$. If an image has already been processed it is possible to cache parts of the processed information and give an answer in an average time of $1ms$.

3.3 Object Color

In order to evaluate the quality of the colors extracted by our system, we use the *color-texture* dataset. We use a metric for the evaluation of the extracted colors which takes into account the amount of color in relation to the quantity of color labeled, as explained in [10]. Since the computational time of the algorithm depends on the size of the image, we have investigated the way to reduce it, resizing the images. We have thus reached a tradeoff between speed and quality of results by resizing the images to 50×50 pixels. The average time required to extract principal colors is $37ms$, obtaining the following performance: Precision: 55%, Recall: 65%, Overall Accuracy: 61%.

Table 1: Training and test results for all datasets used. Each class of objects was assessed individually by measuring the object (Obj) and background (Bg) accuracy, using the VOC2010 [9] evaluation metric.

Dataset	**Obj Train -**	**Bg Train -**	**Obj Test -**	**Bg Test**
Underwear	73%	91%	68%	88%
Lingerie	67%	91%	64%	91%
Menswear	79%	83%	73%	79%
Womenswear	64%	85%	57%	83%
Tie	84%	96%	75%	93%
Shoe	81%	94%	69%	87%
Hat	78%	91%	76%	88%

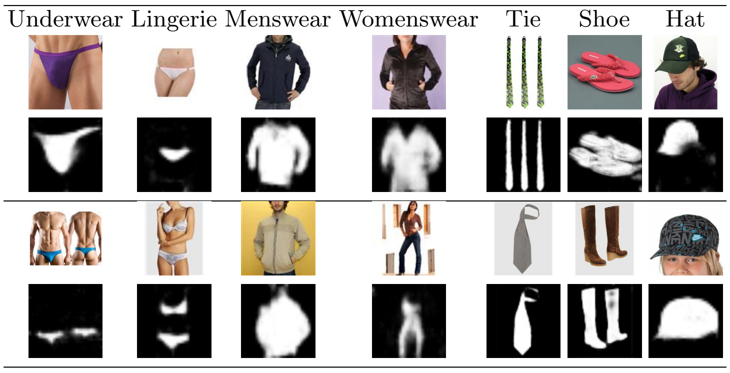


Fig. 7: In each column two examples of input images and segmented objects, belonging to the datasets used in this work. All the images belong to the validation set.

3.4 Object texture

Table 2 shows the confusion matrix obtained evaluating the texture classification method, which can be summarized in an overall accuracy of 75.8% and a K coefficient of 0.73, resulting in a substantial agreement between the truth data and the classifier predictions [11]. Experiments show that the chosen features have a good discrimination power on the used texture classes.

Using 200×200 pixels images, the average time required to extract texture features and predict texture class is $73ms$, where almost all of the time is spent in the feature extraction phase.

4 Conclusions

We created a visual search engine for shoppers based on a new neural algorithm for object segmentation. Starting from the segmented objects we extracted and indexed information on color and texture, transforming all unit of information extracted from images in text descriptions.

Table 2: Confusion matrix for the texture classification problem.

	horiz	**leopard**	**other**	**polka**	**rhombus**	**color**	**squares**	**vertic**	**zebra**	**PA**
horiz	19	0	0	0	0	0	0	0	2	91%
leopard	0	14	4	4	0	0	1	0	1	58%
other	0	2	13	7	2	0	4	0	2	43%
polka	0	0	0	10	0	0	0	0	0	100%
rhombus	0	0	1	1	13	0	0	0	2	77%
color	4	0	2	0	0	37	1	0	0	84%
squares	0	0	0	0	0	0	14	0	0	100%
vertic	0	0	2	0	0	3	0	22	0	82%
zebra	2	0	2	1	0	0	0	0	15	75%
UA	76%	88%	54%	44%	87%	93%	70%	100%	68%	

As demonstrated from the experimental phase and by the application developed, the proposed method is very efficient and can be compared in performance to a text indexing engine.

In future works we want experiment with the extraction and indexing of new visual information from images and improve the performance of the segmentation algorithm in order to obtain more accurate details for the segmented objects.

References

1. Wang, J.Z., Li, J., Wiederhold, G.: Simplicity: Semantics-sensitive integrated matching for picture libraries. IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001) 947–963
2. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recognition **40** (2007) 262–282
3. Skopal, T.: Where are you heading, metric access methods?: a provocative survey. In: Proc. of SISAP 2010. SISAP 2010, New York, NY, USA, ACM (2010) 13–21
4. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning **20** (1995) 273–297
5. Gallo, I., Nodari, A.: Learning object detection using multiple neural netwoks. In: VISAP 2011, INSTICC Press (2011)
6. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE conf. on Neural Networks. (1993) 586–591
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR. (2005) 886–893
8. Nodari, A., Gallo, I., Cavallaro, A.: Real time color image indexing using a textual approach. In: Submitted to *ICIP'11*. (2011) Submitted to *ICIP'11*.
9. Everingham, M., Van Gool, L., e.a.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88** (2010) 303–338
10. Pedoia, V., Colli, V., e.a.: fMRI analysis software tools: an evaluation framework. In: SPIE Medical Imaging 2011. (2011)
11. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33** (1977) 159–174