Hair detection, segmentation, and hairstyle classification in the wild

Article *in* Image and Vision Computing · February 2018

DOI: 10.1016/j.imavis.2018.02.001

4 authors:

Umar Riaz Muhammad
University of Surrey
5 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Michele Svanera
University of Glasgow
15 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Riccardo Leonardi
Università degli Studi di Brescia
263 PUBLICATIONS **2,497** CITATIONS
SEE PROFILE

Sergio Benini
Università degli Studi di Brescia
62 PUBLICATIONS **503** CITATIONS
SEE PROFILE

Image and Vision Computing 71 (2018) 25–37

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Hair detection, segmentation, and hairstyle classification in the wild☆

Umar Riaz Muhammad[a,b], Michele Svanera[a,c,*], Riccardo Leonardi[a], Sergio Benini[a]

[a] Department of Information Engineering, University of Brescia, Brescia, Italy
[b] School of EECS, Queen Mary University of London,London, UK
[c] Institute of Neuroscience and Psychology, University of Glasgow, Glasgow, UK

ARTICLE INFO

Article history:
Received 28 June 2017
Received in revised form 22 January 2018
Accepted 4 February 2018
Available online 15 February 2018

Keywords:
Hair detection
Hair segmentation
Hairstyle classification
Texture analysis
Hair database

ABSTRACT

Hair highly characterises human appearance. Hair detection in images is useful for many applications, such as face and gender recognition, video surveillance, and hair modelling. We tackle the problem of hair analysis (detection, segmentation, and hairstyle classification) from unconstrained view by relying only on textures, without a-priori information on head shape and location, nor using body-part classifiers. We first build a hair probability map by classifying overlapping patches described by features extracted from a CNN, using Random Forest. Then modelling hair (resp. non-hair) from high (resp. low) probability regions, we segment at pixel level uncertain areas by using LTP features and SVM. For the experiments we extend *Figaro*, an image database for hair detection to *Figaro1k*, a new version with more than 1000 manually annotated images. Achieved segmentation accuracy (around 90%) is superior to known state-of-the-art. Images are eventually classified into hairstyle classes: straight, wavy, curly, kinky, braids, dreadlocks, and short.

1. Introduction

Over the course of time, computer vision tasks related to human analysis, such as face detection and recognition, or pedestrian detection and tracking, have always received high research attention. Recent advances allowed the development of robust algorithms able to perform human detection in quite unconstrained conditions, usually referred to as "in the wild", such as the detection method for faces from unconstrained views proposed in [1]. Despite this progress there are still some loopholes or rather some room to increase the robustness of existing human detection algorithms.

One such limitation comes from the incapability of most of the existing methods to detect human presence in fully unconstrained conditions. More specifically we are referring to those cases when we need to detect human presence from the back or over-the-shoulder views, with no clear head-and-shoulder profile, and hence when the only available information is human hair.

Apart from this specific detection problem, which can be useful for surveillance applications [2], head detection tasks [3], facial part segmentation [4], and shot type analysis in movies [5, 6], hair analysis also plays an important role for face recognition [7] and gender classification [8, 9]. In fact many studies confirm that hair is the most important single feature for recognizing familiar faces [10, 11, 12, 13], especially for recognizing faces of your own gender [14].

Last, hair segmentation is a prior step for enabling interesting consumer applications, including 3D hair modelling from a single portrait image [15], portrait pop-ups and hairstyle virtual tryon [16], virtual hair cutting and physical hair simulation [17], and portrait relighting and 3D-printed portrait reliefs [18].

In spite of its importance, hair is rarely analysed, probably because of various problems that make this task particularly challenging. These problems include: wide number of variations in hairstyle, colour and dye; non-rigid structure which may vary depending on ambient conditions, such as wind; variations in visual appearance depending on the head-tilt angle; possible presence of complex backgrounds. To overcome such obstacles, most approaches in literature make assumptions that restrict the applicability of hair analysis. Typically these include the presence of a plain background [7], faces depicted in near-frontal view [7, 19, 20, 21, 22, 23] or a-priori known spatial hair information [19, 20, 21, 24, 25, 26, 27].

Another limitation which probably prevented research on this topic is the scarcity of shared databases annotated for hair detection from unconstrained views. Most repositories for facial features [20, 28, 29] contain only frontal or near-frontal views and are normalized according to face or head size. With the exception of

☆ This paper has been recommended for acceptance by Georgios Tzimiropoulos.

* Corresponding author.
E-mail addresses: u.muhammad@qmul.ac.uk (U.R. Muhammad), Michele.Svanera@glasgow.ac.uk (M. Svanera), riccardo.leonardi@unibs.it (R. Leonardi), sergio.benini@ing.unibs.it (S. Benini).

https://doi.org/10.1016/j.imavis.2018.02.001
0262-8856/

OpenSurfaces [30], which has never been employed in any study for hair analysis yet and which is lacking further subdivision into hairstyles, no database provides segmented hair masks on images from unconstrained views.

1.1. Paper aims and organization

Motivated by its importance, the work proposed here can be regarded as the first attempt to perform a complete hair analysis (detection, segmentation, and hairstyle classification) from unconstrained view without a-priori knowledge on body parts, thus without prior face or head-and-shoulder detection.

The adopted texture-based approach is motivated in order to overcome the typical obstacles of hair analysis. Opting for a coarse-to-fine approach, we start off by producing a hair probability map generated by a binary classification of overlapping image patches represented by texture descriptors. By learning models from texture information derived from high-probability hair regions (and from high-probability non-hair regions, respectively) we are able to obtain a fine-level hair segmentation on uncertain image areas. For performing texture analysis we here extend our initial work on hair analysis in [31] where Linear Ternary Pattern (LTP) [32] features have proven to provide promising results. Inspired by the cross domain success of Convolutional Neural Networks (CNNs) [33], which are recently employed also for segmentation purposes [34], and the potential of various layer outputs as texture descriptors [35], we here consider *deep features*, i.e. descriptors obtained by truncating different CNNs (specifically, CaffeNet [36] and VGG-VD[37]) at different layers.

Even if attitudes towards hairstyles or hair removal may vary widely across different cultures and historical periods, hairstyle is one of the defining characteristics of humans, often related to person's social position, such as age, gender, religion [38]. Despite the fact that human can drastically manipulate their hair, they typically do not [7], so that hair appearance and attributes may provide useful cues also for the recognition task. Hence, as another contribution, for the first time automatic hairstyle recognition is performed here by means of a multi-class texture classification step on the previously segmented hair region.

As a last contribution, we present and share *Figaro1k*, an extension of *Figaro* [31], a multi-class image database for hair detection in the wild[1]. *Figaro1k* contains 1050 unconstrained view images with persons, subdivided into seven different hairstyle classes (*straight, wavy, curly, kinky, braids, dreadlocks, short*), where each image is provided with the corresponding manually segmented hair mask, acting as a ground-truth. Examples of images and related ground-truth masks are given in Fig. 1, while a description of the database is in Section 4.

This document is organized as follows. In Section 2 we review previous work done on hair analysis and texture analysis. In Section 3 we present the workflow of the processing chain, providing a separate description for each block. In Section 4 we describe the generation process and the main characteristics of *Figaro1k*. In Section 5 we focus on the implementation and experimental results. Conclusion marks and future work are drawn in Section 6.

2. Previous work

In this section we give a brief summary of the previous work on hair analysis, along with the approaches that have been used for texture analysis.

[1] Download the database from http://projects.i-ctm.eu/en/project/figaro

2.1. Hair analysis

The work of Liu et al. [39] is one of the first dealing with the hair detection problem, where hair regions are segmented using a-priori spatial and geometrical information. After this initial effort, the hair detection problem remains in the background of computer vision research until the work of Yacoob et al. [7]. Assuming a face in frontal view, the authors propose an algorithm for automatic detection of hair which consisted of face and eye detection followed by skin and hair colour modelling. Rousset et al. [20] propose a hair segmentation algorithm based on a matting technique. First information from frequential and colour analysis is extracted in order to create binary masks as descriptors of hair location. Then a matting method is applied to get the final hair mask. Lee et al. [19] instead introduce a probabilistic approach to perform hair segmentation. They use a Markov Random Field approach and optimize it by extending segmentation algorithms such as Graph-Cut [40] and Loopy Belief Propagation [41] for hair and face segmentation. Roth and Liu [22] propose a learned hair matcher using shape, colour, and texture features derived from localized patches through an AdaBoost technique, which they apply on a subset of images from LFW funneled dataset [29] where all face regions are reasonably aligned with the in-plane rotation (roll) removed.

All the above methods suffer by two principal limitations: they require near-frontal face and/or a body part detector in order to perform hair detection. The first attempt to overcome the limitation of frontal face view is the work by Wang et al. [24]. They use a Bayesian based method for hair segmentation, which shows tolerance to small pose variations. Later on, they extend their work [25, 26] to be compatible with multi-pose situations, including over-the-shoulder view, by proposing a coarse-to-fine hair segmentation method based on a combination of probability maps, fixation points and graph-based segmentation. In the work of Wang et al. [27] the authors propose a method for hair segmentation which does not require the initial detection of the face and/or eyes. First background subtraction is performed to obtain foreground objects, and then human head is detected with a trained human head detector. Next, hair segmentation is carried out using K-mean clustering. While these recent approaches overcame the limitation of frontal view, they still employed body part detectors to help the task of spatially locating hair.

2.2. Texture analysis

Texture segmentation and classification have been interesting for various researches for quite a while, dating back to the work of Laws [42]. In 1991 Jain et al. [43] present an unsupervised texture segmentation algorithm that uses a fixed set of Gabor filters. Using *textons* as frequently co-occurring combinations of oriented linear filter outputs, Malik et al. [44] propose texton learning using a K-means approach. Later on Malik along with Leung [45] studies the recognition of surfaces made from different materials such as concrete, rug, marble, or leather on the basis of their textural appearance. They propose an approach based on building a universal texton vocabulary that could describe generic local features of texture surfaces. Varma and Zisserman [46] show that the texture classification problem can also be tackled effectively by employing only local neighbourhood distributions without the use of large filter banks, which had been done in majority of the works till that point. Other significant contributions include the work by Caputo et al. [47], Ferrari and Zisserman [48], Schwartz and Nishino [49], and Sharan et al. [50].

Since the breakthrough by Krizhevsky et al. [9] lots of computer vision tasks have been improved by the use of Convolutional Neural Networks (CNNs). Texture analysis is no exception. The work of Cimpoi et al. in [35, 51, 52] represents the current state-of-the-art:

Fig. 1. *Figaro1k* contains 1050 hair images belonging to seven different hairstyles (from left to right: straight, wavy, curly, kinky, braids, dreadlocks, short). First row: original images (one per hairstyle class). Second row: related ground-truth masks (in blue).

the authors obtain local texture descriptors by truncating a CNN (VGG at *conv5*) and then use Improved Fisher Vector [53] as a pooling strategy, in opposition to traditional fully connected *(fc)* pooling.

3. Proposed method

In this section we describe in detail the methodological basis of the proposed processing chain to achieve hair detection, segmentation and hairstyle classification in the wild. The overall visual workflow is depicted in Fig. 2.

3.1. Hair detection on image patches

The first detection step aims at creating a probability map of hair presence at patch level, to help in distinguishing image patches which likely contain hair from those belonging to the background. We tackle this task by a classification pipeline which is purely based on hair texture analysis. As shown in Fig. 3 detection involves: a) a feature extraction phase, where hair and non-hair patches from the training set are represented by texture descriptors; b) a model learning phase, where texture descriptors representing hair and non-hair classes are exploited by a machine learning method to train a classifier; and c) the final classification phase, where the trained classifier eventually categorizes patches of the input image either as hair or non-hair.

This final phase of hair detection uses a moving window which classifies overlapping square patches, where the amount of overlap is controlled by the step size of the moving window. Overlapping patches are adopted in order to have multiple classifications on single pixels. This procedure generates a probability map of hair presence in the image, where pixels classified most times as hair indicate the regions with the highest probability of containing hair. Conversely pixels which are never, or almost never, classified as hair represent the regions with the least probability of being hair.

As documented in detail in Section 5.2, several experiments are carried out to produce hair probability maps, by comparing different texture features (LTP, CaffeNet, and VGG-VD), by augmenting the training set, and by adopting different solutions for processing border patches in order to have all pixels classified the same amount

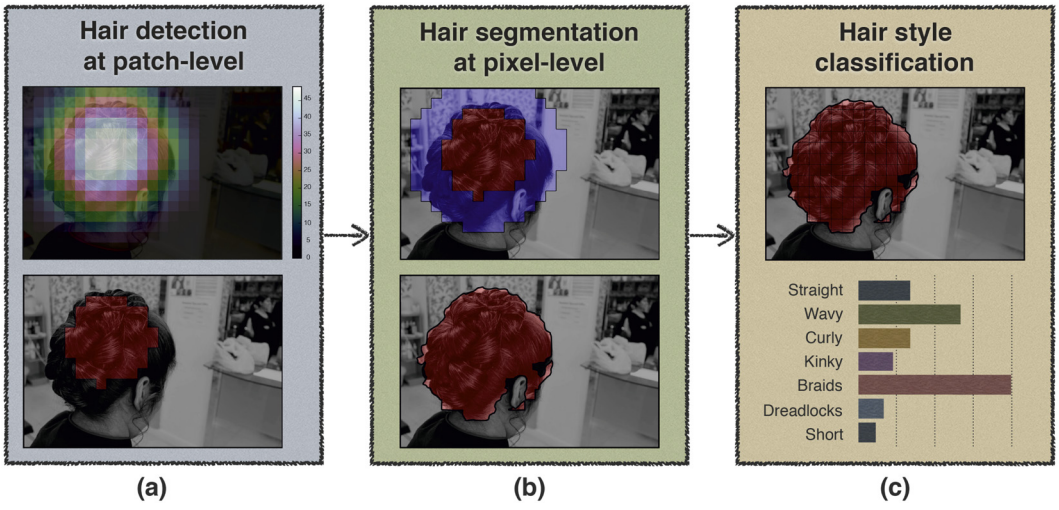


Fig. 2. Logical workflow of the proposed processing chain: a) a hair probability map (top) is created on image patches for identifying high probability hair (white) vs. non-hair (black) regions: as a result a high-probability hair region is returned (bottom, in red); b) hair segmentation is achieved by classifying image pixels in the uncertain area only (in blue, top) thus refining the final hair region (bottom, in red); c) hairstyle is classified by adopting a majority voting scheme on the previously segmented hair patches.

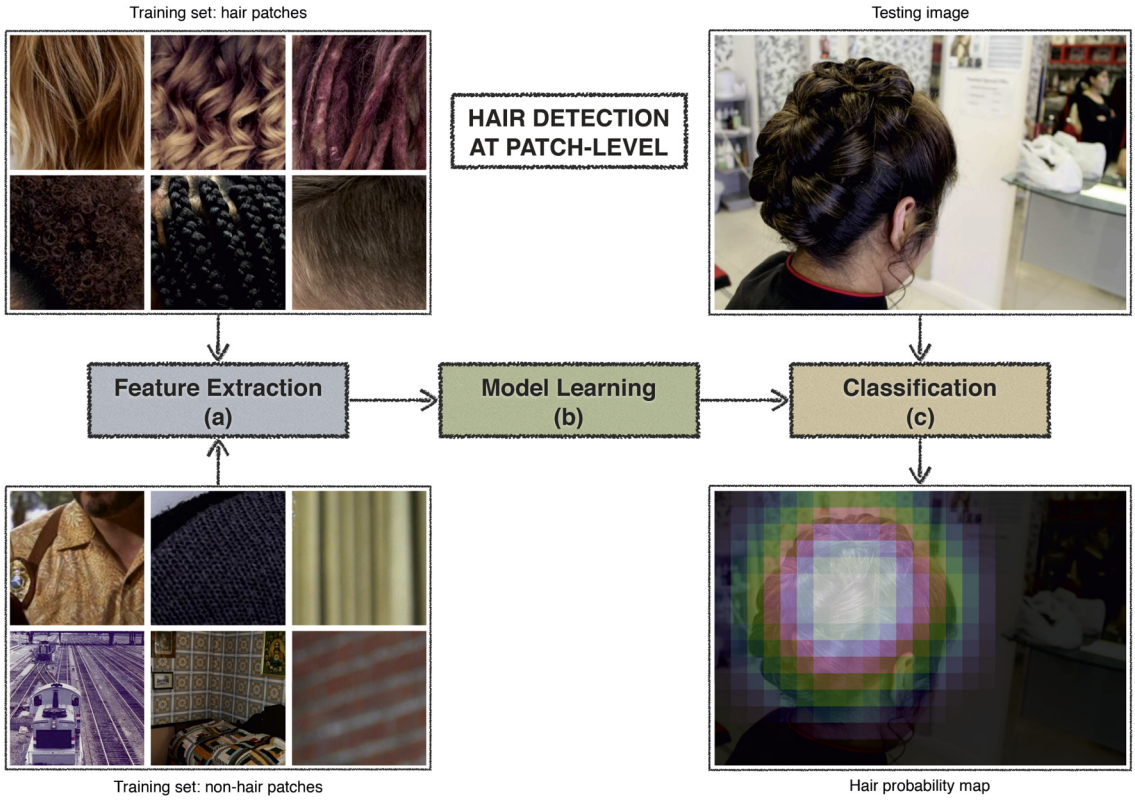


Fig. 3. Hair detection at patch level: a) feature extraction from pure hair vs. non-hair patches; b) model learning; c) classification step on overlapping patches of the test image.

of times. During the experiments the best setting for generating the hair probability map is chosen by fixing a high precision value and choosing the configuration which returns the highest recall on the ground truth masks. The same values of precision and recall are then used to get the probability which best identifies the hair region. A similar procedure is then applied to non-hair regions to isolate background patches. As presented in Section 5.2 best results are obtained by a RF classifier fed with CaffeNet-fc7 features, average colour handling and no data augmentation.

3.2. Hair segmentation on image pixels

The second step aims at segmenting hair at pixel level. Having previously identified (highly probable) hair and background patches, segmentation is performed only where the hair presence is uncertain, as shown in Fig. 4c.

To do this we exploit the texture information present in the already detected hair and non-hair regions of the currently processed image. This means that while hair and non-hair models used

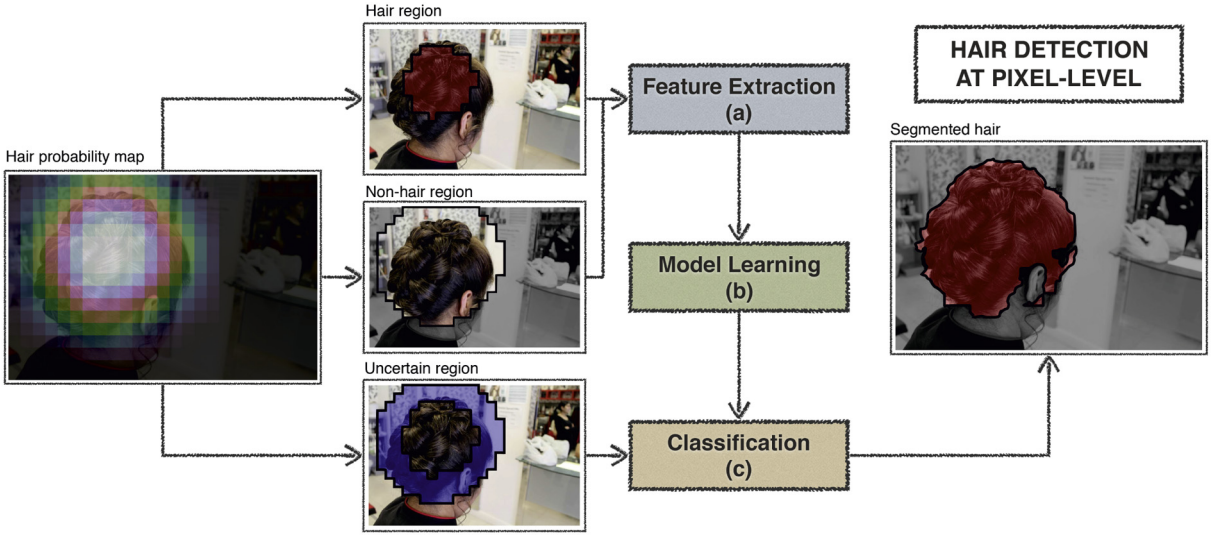


Fig. 4. Hair segmentation at pixel level: a) feature extraction from high-probability hair (in red) vs. non-hair regions (in grey); b) representation of hair vs. non-hair models of the specific image; c) segmentation is performed as a patch classification step (central pixel classification) only on the uncertain region (in blue).

for patch-level analysis are generated by considering the whole training set of images, for classifying hair pixel in the uncertain areas of an image, we rely on the hair and background information which are specific of the image itself. The procedure is shown in Fig. 4.

Keeping in mind that using a single image we have limited data available for training a classifier, feature extraction is performed on overlapping patches obviously smaller than those adopted during the hair detection at patch level. The fine level hair segmentation is achieved by adopting a central pixel labelling scheme, where the result of patch classification is assigned to the central pixel. Adopted texture features for classification, patch size, configuration parameters, performed experiments, and segmentation performance computed against the manually annotated masks are documented in detail in Section 5.3. Best results are obtained by using LTP features and a SVM classifier.

3.3. Hair classification

In this last step of the processing chain, we identify the hairstyle of the depicted person in the image, by choosing among the seven balanced style classes present in *Figaro1k*: straight, wavy, curly, kinky, braids, dreadlocks, or short. The learned classifier performs multi-class classification on hair patches extracted from the segmented hair region obtained from the previous step. For the final hairstyle labelling we adopt a voting scheme, the class receiving the most votes being considered the final label, as shown in Fig. 2c. All performed settings, experiments, and results on hairstyle classification are described in Section 5.4. Best results are obtained by using LTP features and a RF classifier.

4. Databases

In order to carry out hair analysis in the wild, a database with unconstrained view images containing various hair textures is needed. Most repositories available for facial features, including hair information, such as [7, 20, 22, 28] contain only frontal or near-frontal views and are normalized according to face or head size. Conversely other repositories which do contain variations in head-shoulder poses and background [15] [54] are not publicly available. Authors of [25, 26, 55, 56] contacted via mail for sharing the annotated databases never answered, so it was not possible to adopt them for comparison. The scarcity of such databases pushed us to build and share *Figaro1k* (extension of *Figaro* [31]), an annotated multi-class image database for hair analysis in the wild.

4.1. Figaro1k description

Hair exists in a variety of textures, depending on curl pattern, volume, and consistency. We adopted seven possible hairstyle classes, namely *straight, wavy, curly, kinky, braids, dreadlocks and short*. This categorization extends the four types in Walker's taxonomy [57] (*straight, wavy, curly, kinky*) by three categories (*braids, dreadlocks, short*), which broaden the scope of the analysis.

We collected images from Google Images and Flickr, restricting our choice mostly on content in creative commons. When choosing images we adopted some strict guidelines: a significant visible presence of hair texture, balanced distribution of images on all hairstyle classes, unconstrained view images with a particular focus on over-the-shoulder view, balanced number of male and female subjects, different levels of background complexity, diverse hair colours, hairstyle variations for each class on hair arrangements and length. With these criteria in mind, we manually filtered the retrieved images to remove errors and unsuitable samples, to finally obtain a database of 1050 images, 150 for each class.

Being images different in size and aspect ratio, a normalization procedure has been applied. Since individuals are shot from all angles, it was not possible to carry out typical normalization procedures used for frontal views, such as a fixed pixel distance between eyes. The employed normalization factor is chosen so as to reduce the size of the maximum square area inscribed in the hair region to 227×227 pixels, which is the minimum patch size required for further deep feature extraction. As a consequence, images with hair regions smaller than 227×227 pixels have been discarded. This procedure does not limit the applicability of the method to smaller images, as we show in the multi-scale analysis in Section 5.2.4. Eventually, since most images contain hair in a relatively small region if compared to the full image size, we cropped each spatial image dimension to a maximum of 1000 pixels. As a result, the average image size over the database is 718×635 (height, width). Regarding the relative size of the hair w.r.t. the images, since the hair region contains at least 227×227 pixels, hair presence covers on average the 11% of the normalized image.

For each normalized image the ground truth of hair regions has been manually labelled at pixel level using an image analysis tool. As a result, a binary mask indicates the hair location in each image as in Fig. 1 (blue for hair, grey for non-hair).

4.1.1. Patch-F1k

Patch-F1k is an auxiliary database used only for training hair detection at patch level. It is publicly shared and contains 1050 pure hair texture images (227×227) and 1050 pure non-hair texture

Fig. 5. First row: *Patch-F1k* hair patch examples from the corresponding *Figaro1k* examples in Fig. 1. Second row: *Patch-F1k* non-hair patch examples.

images (same size), for a total of 2100 images. Hair patches of *Patch-F1k* are directly extracted from images of *Figaro1k* (first row of Fig. 5), while non-hair patches are partially extracted from non-hair regions in *Figaro1k* pictures (420 images), and partially from VOC2012 [58] (again 420), for a total of 840 non-hair images. The reason of having these non-hair samples taken from VOC2012 is to have a wider variety of backgrounds, so as to make the first level coarse hair detection step more robust, as detailed in Section 5.2.

5. Experimental results

In this section we describe the experimental details of the proposed processing chain with the corresponding results. Since we tackle hair detection, segmentation and hairstyle classification by adopting a pure texture analysis approach, we first describe the employed texture descriptors.

5.1. Texture descriptors

Considered descriptors are Local Ternary Patterns [32] and deep features.

5.1.1. Local Ternary Pattern features

Proposed as a generalization of the Local Binary Pattern (LBP) [59], LTP can be considered as a way of summarizing local grey-level structure. Given a local neighbourhood around each pixel, LTP thresholds the neighbour pixels at the value of the central one (± a threshold *LTP_thr*) and uses the resulting binary-valued image patches as local descriptors. To reduce the number of bins in the histogram forming the final feature, we adopt the uniform pattern extension of the algorithm [32], which assigns all non-uniform patterns to a single bin, obtaining a feature of 118 dimensions.

5.1.2. Deep features

Deep features are obtained using the transfer learning approach on Convolutional Neural Networks (CNNs). The overall process consists of training a deep model on a very large database, and then using the learned model as a feature extractor to obtain descriptors. In our implementation for the extraction of deep features we use CaffeNet model [36], which is a replication of AlexNet model [60] with some minor differences in its default hyper-parameters, and VGG-VD model [37], which has already proven to be an efficient deep feature extractor for local texture description [35]. Our implementation differs from the one presented in [35], since we are not interested in the local texture descriptors but rather in the texture represented in the whole extracted patch.

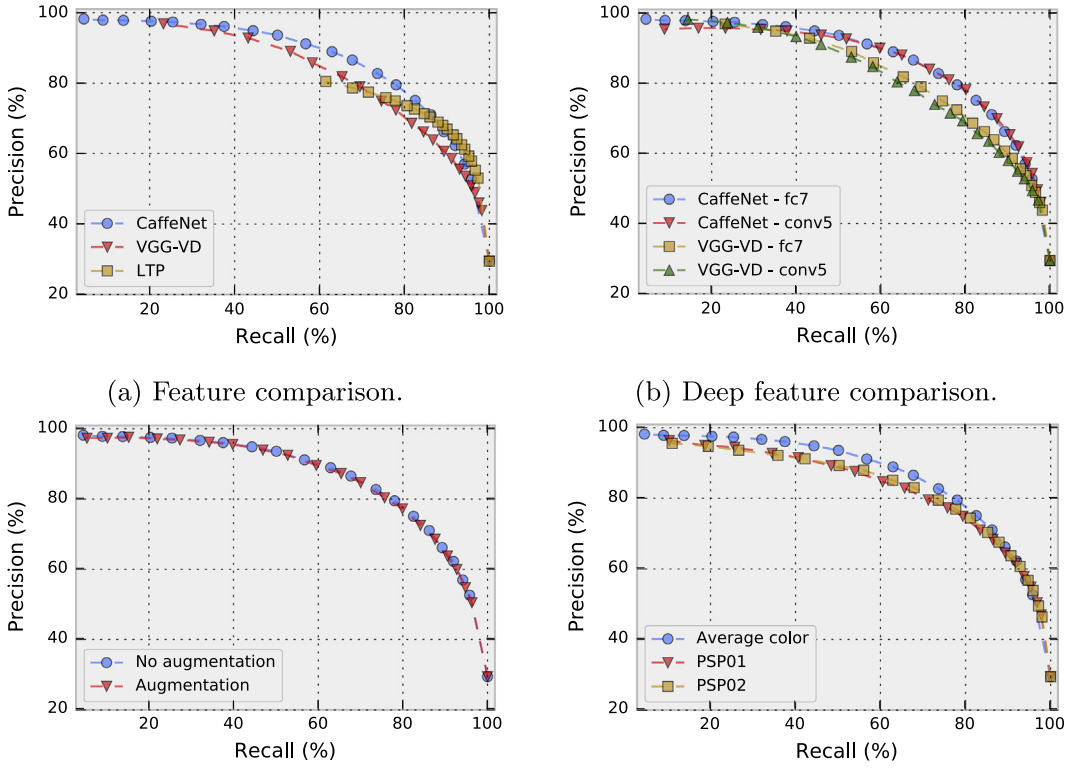
For CaffeNet deep features the considered patch is 231×231 pixels (only the central 227 × 227 portion is taken as input), while for VGG-VD the patch is 224 × 224. As described in Section 5.2.3 we compare the performance of deep features extracted across several layers. In order to make the feature dimensions compatible, we apply Principal Component Analysis (PCA) to reduce the huge dimensionality of lower convolutional layers and obtain a feature length of 4096, while for *fc* layers we take the raw output as they are already 4096 units long.

5.2. Hair detection

In order to perform the proposed hair detection pipeline, we first prepare data and select the optimal parameters for the employed texture descriptors.

5.2.1. Data preparation and parameter selection

A testing set of 210 images randomly chosen from *Patch-F1k* is used through the first experiments for selecting best parameters and



(a) Feature comparison.

(b) Deep feature comparison.

(c) Data augmentation.

(d) Border handling comparison.

Fig. 6. Tests carried out for performance evaluation and comparison.

comparing performance of different approaches. For training we use 840 pure hair and 840 pure non-hair texture patches from *Patch-F1k* (see Fig. 5), ensuring that they are not contained in the test set. A k-fold ($k = 5$) cross-validation is eventually performed (in Section 5.3.3) only on the best approach to ensure its generalisation.

Optimal parameters for LTP feature are selected based on the study in [31]. This work compares different combinations of LTP and HOG [61] parameters on hair detection using the first *Figaro* database, concluding that LTP with patch size 35 × 35 and the threshold value $LTP\_thr = 0.02$ provide the best results. Regarding deep features, we adopt CaffeNet and VGG-VD models pre-trained on ImageNet ILSVRC [62] data, and we compare feature vectors extracted from various convolutional layers and from fully connected ones.

5.2.2. Classification parameters

We train the classifier on the descriptors extracted from the patches using Random Forest (RF) [63], an approach which adopts an ensembling strategy of combining multiple weak classifiers to obtain a final strong classifier.

Classification is performed on a square moving window whose dimension is equal to the patch size required by each texture descriptor. The step size of the moving window is chosen so that the number of times a pixel is classified is the same for all descriptors, to obtain comparable probability maps. Hence for LTP descriptor the patch is 35 × 35 and the step size 5 in each dimension, for CaffeNet based deep feature the patch is 227 × 227 with step size 33 in each dimension, and for VGG-VD based features the patch is 224 × 224 and the step size 32. The probability maps generated with the aforementioned parameters range over 50 different values for all texture descriptors, i.e. a pixel can be classified as hair from 0 times (minimum value) up to 49 times (maximum value).

5.2.3. Results on Figaro1k

The description of the analyses performed in order to obtain the best performance follows. Tests are executed in a progressive way, each step highlighting the best choice on a subset of overall parameters. Performance are evaluated by considering precision and recall values obtained by thresholding the probability maps, i.e. the classification output, at different levels (between 0 and 100% with the step of 5%) and comparing them with the ground truth hair masks.

We start off by comparing the performance of LTP features versus deep features as shown in Fig. 6a: deep features (both CaffeNet and VGG-VD) outperforms LTP features which were the former best hair descriptors in [31].

We then proceed by analysing the performance of the two deep features obtained by using the output of various convolutional (*conv*) layers and fully connected (*fc*) layers. Among *conv* layers, *conv5* based deep features performs the best, while among *fc* layers *fc7* gives the best performance. Overall CaffeNet-fc7 outperforms all other deep features, as shown in Fig. 6b (where only the best *conv* and *fc* layers are presented, for the sake of clarity).

After selecting the best deep feature, we try to see whether classical data augmentation strategies applied on training data improves the results. For data augmentation each input hair/non-hair patch first is rotated left and right by a random quantity (up to a limit of 90°), and then the rotated patch is passed through a low pass filter with probability $p = 0.5$. Second, we generate new data by applying feature standardization of pixel values across the entire dataset. Third a whitening transform on patches is also applied, to better highlight the structures and features in the image to the learning algorithm. Despite the obtained augmented dataset, as reported in Fig. 6c we observe no substantial increase in performance, hence we discard data augmentation from further processing.

It is worth mentioning that the minimum patch size required for deep feature extraction is of considerable dimensions, i.e. 227 × 227, thus the choice of border type becomes of particular relevance. We experiment different approaches to border handling, by comparing average image colour border and two different types of phase scrambling permutation borders, finding out that average image colour border gives the best performance, as shown in Fig. 6d.

5.2.4. Scale invariance evaluation

The following experiment aims at demonstrating that the proposed approach overcomes the problem of detecting hair in images of small dimensions, or whenever the relative size of hair area with respect to the image size becomes small.

Let us imagine that we have an image with a very small hair region. Since to run our method we technically need a minimum patch size of 227 × 227, when we are not detecting any hair region, it might be the case that the image contains a hair region, but this is too small to be detected. To take into account this case, the idea is to up-scale the image using an interpolation filter with increasing factors $L = 2, 3, \ldots$ and apply the proposed detection method. In the case hair is present, whenever the minimum hair patch size reaches at least 227 × 227 pixels (i.e., hair detection is technically feasible) the hair region should be correctly detected.

In order to test this idea, we have first down-scaled 35 randomly chosen images from *Figaro1k* (5 from each hair class) using a decimation filter with factor $M = 4$ (i.e., bringing them to the 25% of the original size), so that hair detection was no longer technically feasible. Then, starting from the down-scaled image, we run the algorithm on up-scaled versions of it using an interpolation filter with factors $L = 2, 3, \ldots, 7$ (i.e., on 50%, 75%, …, 175% of the original size, as shown in Fig. 7), where $L = 4$ corresponds to restoring the original image size (100%). By doing so, if any hair region is present, it will pragmatically reach the minimum patch size enabling detection, even if up-scaled images are clearly at degraded quality with respect to original ones.

As presented in Fig. 8 we can observe that when the scale is below 100% of the original (i.e. with $L < 4$) the algorithm does not detect any hair because hair patches are too small for *technically* being detected. Conversely for all *down- and up-scaled* images above 100%

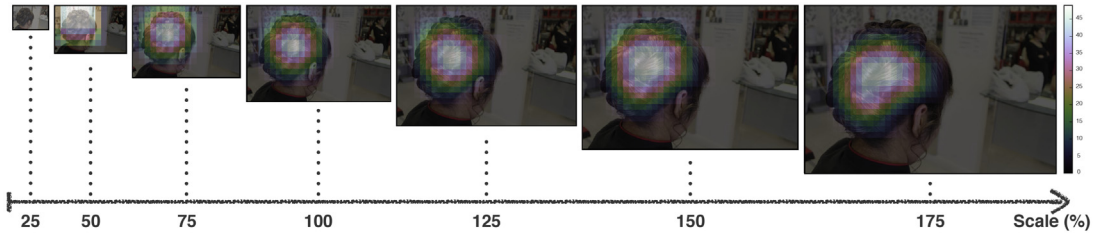


Fig. 7. Hair detection is first performed on a down-sampled image at 25% of the original size, which is afterwards up-scaled at 50%, 75%, 100%, 125%, 150%, 175% (from left to right).

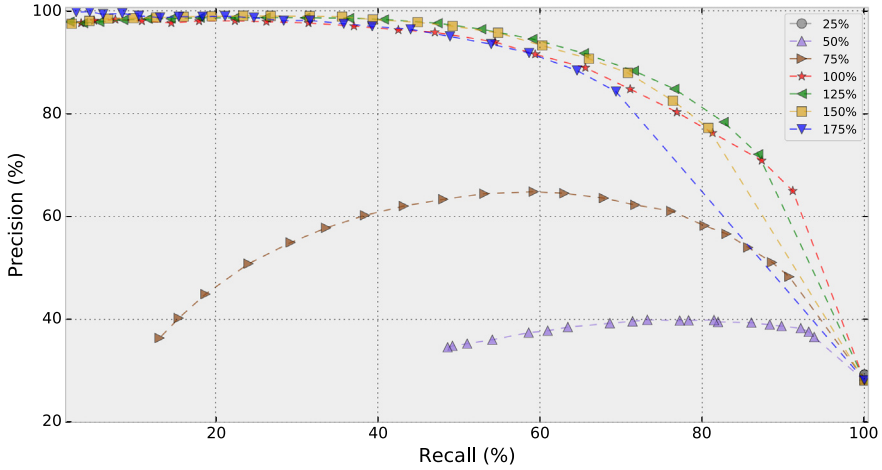


Fig. 8. Test on scale invariance is performed on images which are obtained by up-scaling a down-sampled version at 25% of the original image. The algorithm runs on interpolated versions at 50%, 75%, 100%, 125%, 150%, and 175% of the original size.

(i.e. with $L \geq 4$, that is when the constrain on the minimum hair patch size is satisfied), perform well (and very similarly) even if they are degraded versions of the original ones (because of interpolation), thus showing good robustness of the approach to scale variance and detection of small hair regions.

5.3. Hair segmentation

Hair segmentation is tackled by training a per-image classifier which performs hair segmentation at pixel level.

5.3.1. Data preparation and parameter selection

For training the per-image classifier we exploit the results of the previous step by extracting texture features from image regions where the probability of having hair is higher than 65% of the maximum value of the probability map (measured for each image). This threshold corresponds to the probability level which gives the precision of 95% on hair region detection, as shown in Fig. 9a. We repeat the process for training the recognition of the background by feeding the classifier with texture features extracted from image regions with probability of not having hair higher than 85% of the maximum value of the probability map (measured for each image). This threshold corresponds to the probability level which ensures a precision of 95% on non-hair region, as shown in Fig. 9b. Segmentation is then performed only on the uncertain region (the blue area in Fig. 2b), i.e. where the probability of having hair and non-hair is lower than 95%.

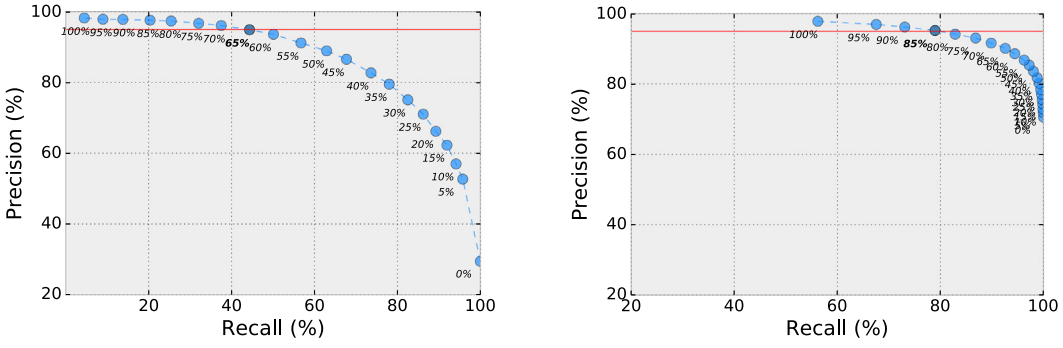
Due to the fine nature of hair segmentation needed for this step, we use only LTP features with patch size of 25 × 25 as the best choice derived from [31], with LTP threshold value $LTP_{thr} = 0.02$, thus discarding the deep features which have performed so well for the coarse level hair detection.

5.3.2. Classification parameters

Best classification results are obtained by using a linear Support Vector Machine (SVM) [64] which we train with overlapping patches of high probability hair and non-hair regions. During the classification of the uncertain region, a moving window extracts overlapping square patches of size 25 × 25, with step size of 3 pixels in each dimension. Such step size is compatible with the adopted central pixel labelling scheme, which labels the result of the patch classification to only the central 3×3 pixels. At the end we refine the obtained result by using a super-pixel representation [65] of the uncertain area and labelling each super-pixel using a majority voting scheme.

5.3.3. Results on Figaro1k

Performance of hair segmentation is first evaluated on the test set of 210 segmented images with respect to the corresponding ground



(a) Hair thresholding. (b) Non-hair thresholding.

Fig. 9. Intersection between the probability curve and the horizontal red line, corresponding to 95% precision, identifies the threshold probability level for hair and non-hair detection.

Table 1
Overall hair segmentation performance (our approaches in bold).

Method (*coarse+fine+segmentation*)	Pr.(%)	Rec.(%)	F_1(%)	Acc.(%)
DeepFeature+LTP+super-pixel	89.0	81.1	84.9	91.5
LTP+LTP+graph-based [31]	81.2	65.1	72.3	85.3
DeepFeature *+(LTP+graph-based* [31]*)*	89.4	69.7	78.3	88.7

Table 2
Hair segmentation performance comparison F_1(%). Performance on *Figaro1k* is computed on the test set of 210 images.

Algorithm	NHD1000	MHD3820	FIGARO	FIGARO1K
[24]	65.6	64.0	–	–
[19]	81.3	66.9	–	–
[25]	85.0	77.3	–	–
[31]	n.a.	n.a.	77.5	72.3
Our method	**n.a.**	**n.a.**	**80.8**	**84.9**

truth masks. Results are shown in Table 1 in terms of precision and recall.

To compare our results we apply on *Figaro1k* the hair segmentation method proposed in [31], where both hair detection and segmentation are achieved using LTP, followed by a graph-based segmentation. To test the specific benefit brought by our segmentation approach we also tried a hybrid algorithm, using the hair detection results obtained by our deep features combined with the hair segmentation method proposed in [31]. All obtained results on *Figaro1k* are reported in Table 1 and are presented for comparison in Fig. 10, where it is evident that the present approach overcomes previous work methodologies.

To ensure the generalisation of the proposed approach, experiments with our best method (DeepFeature+LTP+super-pixel) are repeated on the whole *Figaro1k* dataset using a 5-fold cross-validation. This means that hair detection at patch level is performed by splitting *Figaro1k* in 5 subsets (or folds) of 210 images each. Then we use 4 subsets to train (using the related patches in *Patch-F1k*) and leave the remaining fold as test, and finally average the results against each of the folds. By considering segmentation results at pixel level on the whole *Figaro1k* dataset (i.e. all 1050 images), with respect to results reported in Table 1 we observe a slight increase of average precision ($Pr. = 90.3\%$), a more evident drop in recall ($Rec. = 72.2\%$), which lead to final average $F_1 = 80.0\%$ and $Acc. = 89.5\%$, thus ensuring that the model generalize to new data. End to end k-fold cross-validation on the whole *Figaro1k* dataset is not performed on other methods due to the computational effort required by the extraction of LTP features, as described later in Section 5.3.5.

5.3.4. Comparison across databases

Our method has been trained and tested on the first version of *Figaro* database, where it outperforms the approach in [31], as shown in Table 2. Since other direct comparisons with different methods are not possible due to the unavailability of shared databases (see details in Section 4), we here report performance declared in [25] for indirect comparison. This work compares more methods on a conventional Near-frontal Head-shoulder Database with 1000 images (NHD1000), and a Multi-pose Head-shoulder Database of 3820 images (MHD3820), which also includes 920 non-frontal views. By a simple inspection of images in [25] (the database is not shared) we can estimate that the challenge level provided by the 920 non-frontal views in MHD3820 is comparable with images in *Figaro1k*. However, considering that [25] achieves an overall best value of $F_1 = 77.3\%$ (as reported in Table 2) on MHD3820 images mostly in frontal views (2900 out of 3820), we expect our algorithm which scores $F_1 = 84.9\%$ on *Figaro1k* where most images are in non-frontal view, to be far superior.

5.3.5. Computation time

Training a per-image classifier makes the overall pipeline fairly slow. In particular the bottleneck of the whole process is constituted by the current CPU-based implementation of LTP features, which are extracted from 25 × 25 patches and step size 3 pixels, along each dimension. While the initial hair detection at patch-level (performed with deep features on 227 × 227 patches and step size 33 pixels) is pretty fast (5.31 s on average per image, GPU-based implementation), the extraction of LTP features from high probability hair (resp. non-hair) regions of each image takes an average of 519.94 s per image. Once computed LTP features, the training process of the classifier is pretty fast, 2.18 s on average. After the detection phase, hair segmentation conducted at pixel-level requires the extraction of LTP features from the uncertain region, for an average processing time of 225.56 s. Considering that the post-processing is pretty fast (around 1.79 s per image), the total average processing time per image is 754.78 s. All timings have been computed using an Intel(R) Xeon(R) CPU E5-1650 v4 @3.60 GHz, with 6 cores per socket, a GPU GeForce GTX 1080 Ti, and 64 GB of memory. Computation time could be severely improved by smarter (GPU-based) implementation of LTP features, however in this work we focused more on surpassing other existing methods in terms of classification accuracy, leaving the computational improvements to future work.

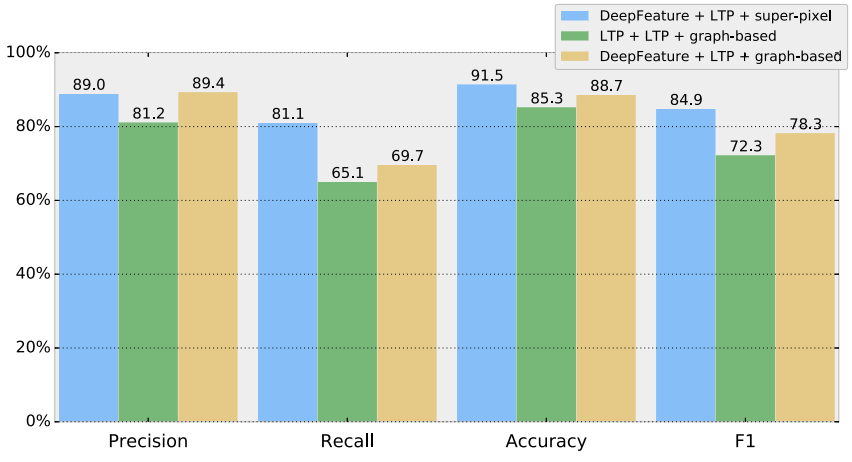


Fig. 10. Overall hair segmentation performance.

Table 3
Hair classification - patch size 25 × 25.

		Predicted labels						
		Straight	Wavy	Curly	Kinky	Braids	Dread.	Short
True labels	Straight	16	4	0	0	0	1	9
	Wavy	6	18	2	0	0	0	4
	Curly	1	1	20	6	0	1	1
	Kinky	1	0	1	25	1	2	0
	Braids	1	6	7	2	10	2	2
	Dread.	1	1	6	1	2	18	1
	Short	2	0	2	6	1	1	18

Table 4
Hair classification - patch size 50 × 50.

		Predicted labels						
		Straight	Wavy	Curly	Kinky	Braids	Dread.	Short
True labels	Straight	18	4	0	0	0	0	8
	Wavy	6	17	5	0	0	0	2
	Curly	0	3	20	4	1	2	0
	Kinky	0	0	3	22	2	3	0
	Braids	1	8	7	1	8	3	2
	Dread.	1	2	8	1	1	16	1
	Short	4	2	3	5	1	0	15

Table 5
Hair classification - patch size 100 × 100.

		Predicted labels						
		Straight	Wavy	Curly	Kinky	Braids	Dread.	Short
True labels	Straight	20	6	0	0	0	0	4
	Wavy	4	20	3	0	1	0	2
	Curly	1	4	17	5	0	2	1
	Kinky	0	0	4	19	3	3	1
	Braids	0	5	10	1	9	2	3
	Dread.	1	3	6	2	2	16	0
	Short	6	4	4	5	1	1	9

5.4. Hair classification

After detection and segmentation, our hair analysis considers the hairstyle.

5.4.1. Data preparation and parameter selection

For training the classifier we use 120 pure hair texture images from *Patch-F1k* for each of the seven types of hair classes (see Fig. 5). Testing is performed on the set of 210 unconstrained view images of *Figaro1k*, 30 for each hairstyle class, which do not contain patches of *Patch-F1k* used for training. Due to the size variety of segmented hair regions, tests are carried out by comparing LTP on three different patch size: 25 × 25, 50 × 50, and 100 × 100, all with $LTP_{thr} = 0.02$.

5.4.2. Classification parameters

Best results are obtained by using a Random Forest (RF) [63], by feeding it with texture descriptors extracted from non-overlapping

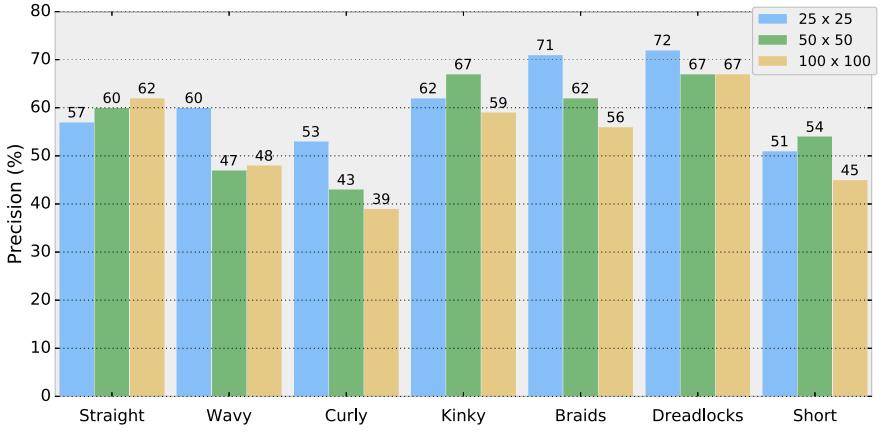


Fig. 11. Precision performance on hairstyle classification.

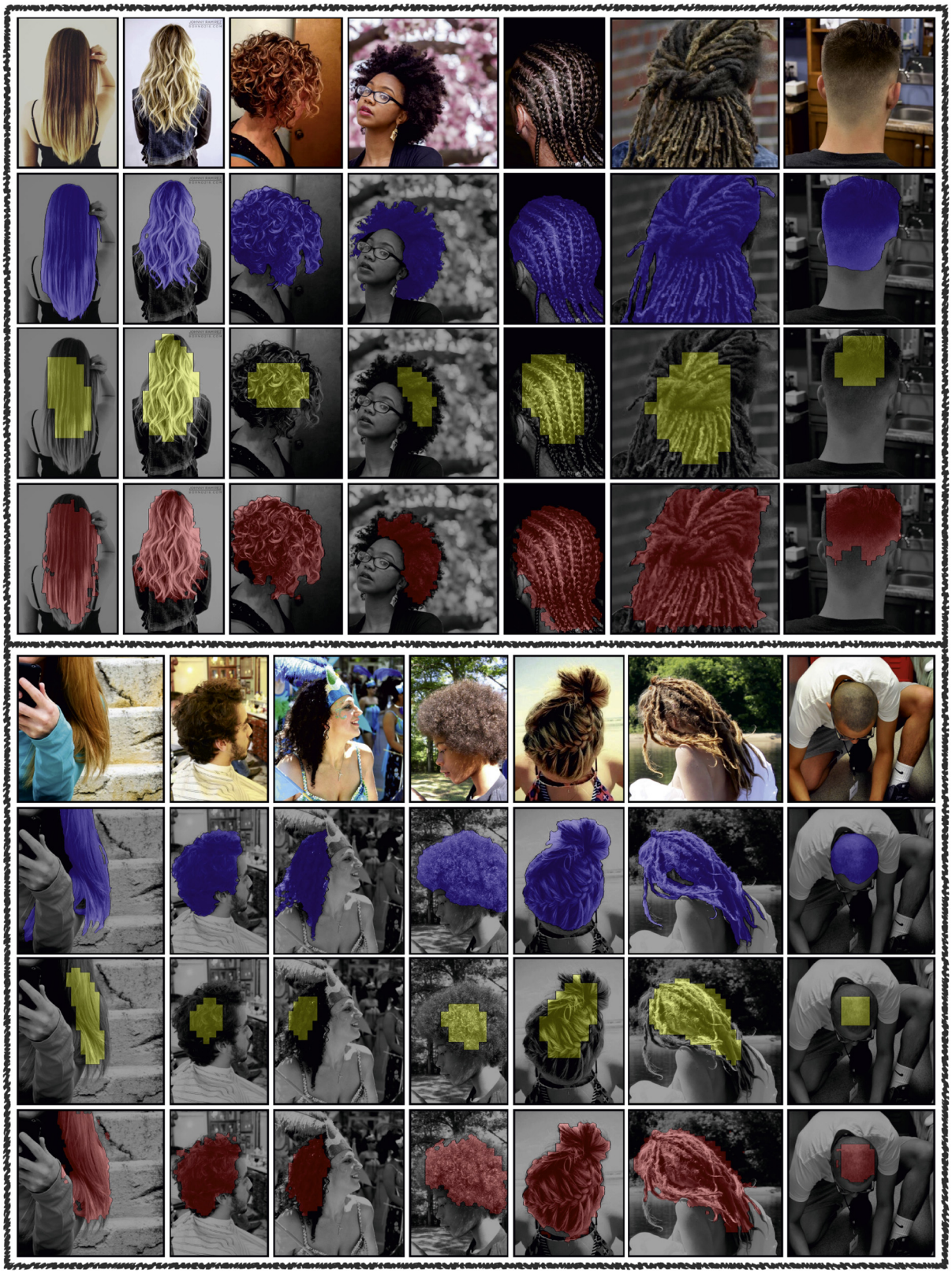


Fig. 12. Fourteen examples (two per hairstyle class) of the proposed method are shown. Displayed outputs for each example: original image (RGB), ground truth (overlay in blue), hair detection method (overlay in yellow), and final segmentation step (overlay in red).

patches of the training set. Classification is performed on a square moving window which extracts non-overlapping square patches of size 35 × 35 from the segmented hair region and classify them into one hairstyle class. The overall hairstyle class is eventually obtained by employing a majority voting scheme on patch classification results.

5.4.3. Results on Figaro1k

Performance of the proposed multi-class hairstyle classification method using patch size of 25 × 25, 50 × 50, and 100 × 100 are shown by the confusion matrices in Tables 3, 4, and 5, respectively. As somehow expected (see Fig. 11) small patches (25 × 25) obtain best precision on highly textured hairstyles (i.e. curly, kinky, dreadlocks, and short), while bigger patches (50 × 50 and 100 × 100) perform best on more regular and less textured classes (i.e. straight and wavy).

6. Conclusion

We perform hair analysis in unconstrained setting, including hair detection, segmentation, and hairstyle classification. The analysis is carried out without a-priori information about head and hair location, nor any body-part classifier, but exploiting only texture features. Using deep features derived from CaffeNet-fc7 and RF classification we first tackle hair detection at patch level. We then refine the obtained results by segmenting hair at pixel level using LTP feature and SVM. Obtained results are superior to other methods exploiting a-priori information on near-frontal face view databases. Some visual results are provided in Fig. 12. As a further contribution, we share *Figaro1k* a database of 1050 annotated hair images taken from unconstrained views, subdivided in seven hairstyle classes, hoping to make a contribution to the problem of hair analysis, and warranting further study on this often-ignored visual cue.

Acknowledgements

Authors U.R. Muhammad, M. Svanera, and S. Benini equally contributed to this research work. The authors would also like to thank B.Sc. students Marco Lombardi and Daniele Baggi from the University of Brescia for their help on database construction and analysis.

References

[1] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE. 2012, pp. 2879–2886.
[2] R. Xu, Y. Guan, Y. Huang, Multiple human detection and tracking based on head detection for real-time video surveillance, Multimed. Tools Appl. 74 (3) (2015) 729–742.
[3] Z. Zhang, H. Gunes, M. Piccardi, An accurate algorithm for head detection based on XYZ and HSV hair and skin color models, 2008 15th IEEE International Conference on Image Processing, IEEE. 2008, pp. 1644–1647.
[4] A.S. Jackson, M. Valstar, G. Tzimiropoulos, A CNN Cascade for Landmark Guided Semantic Part Segmentation, Springer International Publishing, Cham, 2016, 143–155. https://doi.org/10.1007/978-3-319-49409-8_14.
[5] M. Svanera, S. Benini, N. Adami, R. Leonardi, A. Kovács, Over-the-shoulder shot detection in art films, 2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI), IEEE. 2015, pp. 1–6.
[6] S. Benini, M. Svanera, N. Adami, R. Leonardi, A.B. Kovács, Shot scale distribution in art films, Multimed. Tools Appl. (2016) 1–29.
[7] Y. Yacoob, L.S. Davis, Detection and analysis of hair, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1164–1169.
[8] K. Ueki, H. Komatsu, S. Imaizumi, K. Kaneko, N. Sekine, J. Katto, T. Kobayashi, A method of gender classification by integrating facial, hairstyle, and clothing images, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 4, IEEE. 2004, pp. 446–449.
[9] I.A. Kakadiaris, N. Sarafianos, C. Nikou, Show me your body: gender classification from still images, Image Processing (ICIP), 2016 IEEE International Conference on, IEEE. 2016, pp. 3156–3160.
[10] J. Shepherd, H. Ellis, G. Davies, Perceiving and Remembering Faces, Academic Press. 1981.
[11] P. Sinha, T. Poggio, I think I know that face..., Nature (1996)
[12] R.A. Johnston, A.J. Edmonds, Familiar and unfamiliar face recognition: a review, Memory 17 (5) (2009) 577–596.
[13] U. Toseeb, D.R. Keeble, E.J. Bryant, The significance of hair for face recognition, PloS one 7 (3) (2012) e34144.
[14] D.B. Wright, B. Sladden, An own gender bias and the importance of hair in face recognition, Acta Psychol. 114 (1) (2003) 101–114.
[15] M. Chai, T. Shao, H. Wu, Y. Weng, K. Zhou, Autohair: fully automatic hair modeling from a single image, ACM Trans. Graph. (TOG) 35 (4) (2016) 116.
[16] M. Chai, L. Wang, Y. Weng, Y. Yu, B. Guo, K. Zhou, Single-view hair modeling for portrait manipulation, ACM Trans. Graph. (TOG) 31 (4) (2012) 116.
[17] M. Chai, L. Wang, Y. Weng, X. Jin, K. Zhou, Dynamic hair manipulation in images and videos, ACM Trans. Graph. (TOG) 32 (4) (2013) 75.
[18] M. Chai, L. Luo, K. Sunkavalli, N. Carr, S. Hadap, K. Zhou, High-quality hair modeling from a single portrait photo, ACM Trans. Graph. (TOG) 34 (6) (2015) 204.
[19] K.-C. Lee, D. Anguelov, B. Sumengen, S.B. Gokturk, Markov random field models for hair and face segmentation, Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, IEEE. 2008, pp. 1–6.
[20] C. Rousset, P.-Y. Coulon, Frequential and color analysis for hair mask segmentation, 2008 15th IEEE International Conference on Image Processing, IEEE. 2008, pp. 2276–2279.
[21] K. Khan, M. Mauro, R. Leonardi, Multi-class semantic segmentation of faces, Image Processing (ICIP), 2015 IEEE International Conference on, IEEE. 2015, pp. 827–831.
[22] J. Roth, X. Liu, On hair recognition in the wild by machine., AAAI, 2014. pp. 2824–2830.
[23] K. Khan, M. Mauro, P. Migliorati, R. Leonardi, Head pose estimation through multi-class face segmentation, 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017. pp. 175–180. https://doi.org/10.1109/ICME.2017.8019521.
[24] D. Wang, S. Shan, W. Zeng, H. Zhang, X. Chen, A novel two-tier Bayesian based method for hair segmentation, 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE. 2009, pp. 2401–2404.
[25] D. Wang, X. Chai, H. Zhang, H. Chang, W. Zeng, S. Shan, A novel coarse-to-fine hair segmentation method, Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE. 2011, pp. 233–238.
[26] D. Wang, S. Shan, H. Zhang, W. Zeng, X. Chen, Isomorphic manifold inference for hair segmentation, Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE. 2013, pp. 1–6.
[27] Y. Wang, Z. Zhou, E.K. Teoh, B. Su, Human hair segmentation and length detection for human appearance model, Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE. 2014, pp. 450–454.
[28] A.M. Martinez, The AR face database, CVC Technical Report 24 (1998)
[29] G.B. Huang, V. Jain, E. Learned-Miller, Unsupervised joint alignment of complex images, 2007 IEEE 11th International Conference on Computer Vision, IEEE. 2007, pp. 1–8.
[30] S. Bell, P. Upchurch, N. Snavely, K. Bala, OpenSurfaces: a richly annotated catalog of surface appearance, ACM Trans. Graph. (TOG) 32 (4) (2013) 111.
[31] M. Svanera, U.R. Muhammad, R. Leonardi, S. Benini, Figaro, hair detection and segmentation in the wild, Image Processing (ICIP), 2016 IEEE International Conference on, IEEE. 2016, pp. 933–937.
[32] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, International Workshop on Analysis and Modeling of Faces and Gestures, Springer. 2007, pp. 168–182.
[33] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: a deep convolutional activation feature for generic visual recognition., ICML, 2014. pp. 647–655.
[34] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651. https://doi.org/10.1109/TPAMI.2016.2572683.
[35] M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, Int. J. Comput. Vis. 118 (1) (2016) 65–94.
[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM international conference on Multimedia, ACM. 2014, pp. 675–678.
[37] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014.arXiv preprint arXiv:1409.1556.
[38] V. Sherrow, Encyclopedia of Hair: A Cultural History, Greenwood Press. 2006, https://books.google.it/books?id=z_bWAAAAMAAJ.
[39] Z.-Q. Liu, J.Y. Guo, L. Bruton, A knowledge-based system for hair region segmentation, Signal Processing and Its Applications, 1996. ISSPA 96., Fourth International Symposium on, vol. 2, IEEE. 1996, pp. 575–576.
[40] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.
[41] J.S. Yedidia, W.T. Freeman, Y. Weiss, Understanding belief propagation and its generalizations, Explor. artif. intell. new millennium 8 (2003) 236–239.
[42] K.I. Laws, Textured Image Segmentation, Tech. rep., DTIC Document. 1980.
[43] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, Pattern Recogn. 24 (12) (1991) 1167–1186.
[44] J. Malik, S. Belongie, J. Shi, T. Leung, Textons, contours and regions: cue integration in image segmentation, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, IEEE. 1999, pp. 918–925.
[45] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vis. 43 (1) (2001) 29–44.
[46] M. Varma, A. Zisserman, Texture classification: are filter banks necessary? Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on, vol. 2, IEEE. 2003, II–691.
[47] B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2, IEEE. 2005, pp. 1597–1604.

[48] V. Ferrari, A. Zisserman, Learning visual attributes, Advances in Neural Information Processing Systems, 2007. pp. 433–440.
[49] G. Schwartz, K. Nishino, Visual material traits: recognizing per-pixel material context, Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013. pp. 883–890.
[50] L. Sharan, C. Liu, R. Rosenholtz, E.H. Adelson, Recognizing materials using perceptually inspired features, Int. J. Comput. Vis. 103 (3) (2013) 348–371.
[51] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 3606–3613.
[52] M. Cimpoi, S. Maji, A. Vedaldi, Deep filter banks for texture recognition and segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 3828–3836.
[53] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE. 2007, pp. 1–8.
[54] P. Julian, C. Dehais, F. Lauze, V. Charvillat, A. Bartoli, A. Choukroun, Automatic hair detection in the wild, Pattern Recognition (icpr), 2010 20th International Conference on, IEEE. 2010, pp. 4617–4620.
[55] P. Aarabi, Automatic segmentation of hair in images, 2015 IEEE International Symposium on Multimedia (ISM), IEEE. 2015, pp. 69–72.
[56] C. Rousset, P.-Y. Coulon, M. Rombaut, Transferable belief model for hair mask segmentation, 2010 IEEE International Conference on Image Processing, IEEE. 2010, pp. 237–240.
[57] A. Walker, T. Wiltz, Andre Talks Hair!, Simon and Schuster. 1997.
[58] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012), 2012.
[59] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[60] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, 2012. pp. 1097–1105.
[61] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE. 2005, pp. 886–893.
[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE. 2009, pp. 248–255.
[63] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[64] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[65] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.