Expert Systems With Applications 87 (2017) 41–55

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Protecting the privacy of humans in video sequences using a computer vision-based de-identification pipeline

Karla Brkić*, Tomislav Hrkać, Zoran Kalafatić

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia

ARTICLE INFO

Article history:
Received 12 October 2016
Revised 5 May 2017
Accepted 27 May 2017

Keywords:
Privacy protection
De-identification
Computer vision
Video processing

ABSTRACT

We propose a computer vision-based de-identification pipeline that enables automated protection of privacy of humans in video sequences through obfuscating their appearance, while preserving the naturalness and utility of the de-identified data. Our pipeline specifically addresses de-identifying soft and non-biometric features, such as clothing, hair, skin color etc., which often remain recognizable when simpler techniques such as blurring are applied. Assuming a surveillance scenario, we combine background subtraction based on Gaussian mixtures with an improved version of the GrabCut algorithm to find and segment pedestrians. De-identification is performed by altering the appearance of the segmented pedestrians through the neural art algorithm that uses the responses of a deep neural network to render the pedestrian images in a different style. Experimental evaluation is performed both by automated classification and through a user study. Results suggest that the proposed pipeline successfully de-identifies a range of hard and soft biometric and non-biometric identifiers, including face, clothing and hair.

1. Introduction

Surveillance cameras are becoming widespread in public places such as city streets, subway stations, banks, shopping centers, airports etc. Although indispensable in improving personal safety, the ubiquity of video surveillance also raises privacy concerns. A wealth of privacy-sensitive information can be mined on every recorded individual, including for example his whereabouts at a given time of the day, whom he associates with, which bank he uses, which shops he prefers. Given recent advances in computer vision, retrieving this information now requires considerably less effort from a potentially malicious observer (Baltieri, Vezzani, & Cucchiara, 2014; Garcia et al., 2016).

Recognizing the importance of privacy protection, many nations implement strict regulations for governing personal data (see e.g. the Data Protection Directive of the European Union[1]). To be in compliance with such legislation, modern video surveillance systems should aim at minimum information disclosure in accordance with the chain of authority, so that each person able to access privacy-sensitive data is authorized by law for their particular level of access. At the same time, the utility of the data should be preserved throughout the chain of authority. For example, an employee of a video surveillance company should be able to view surveillance footage and assess whether a dangerous situation is occurring, but should not be aware of the identities of all the people in the scene. Ideally, personally identifying features should be obfuscated or removed, while the actions occurring in the scene should still be clearly shown, hence retaining the utility of the data and at the same time protecting the privacy of the filmed individuals. Persons of higher authority, for example police officers, should be able to view the original data with personally identifying features, given that there is a legal justification for them to do so.

In this paper, we consider automated de-identification in surveillance videos based on computer vision. De-identification in images and video sequences is the process of obfuscating the identities of the recorded people in order to protect their privacy. It is achieved by removing or obfuscating various identifying personal features, including hard biometric features such as the face, and soft and non-biometric features such as hair and eye color, gait, body posture, clothing, birthmarks and tattoos etc. (Ribarić, Ariyaeeinia, & Pavešić, 2016). Perhaps one of the most well known and commonly used examples of de-identification is the blurring of faces seen nowadays on services such as Google Street View. Although this method offers a certain level of privacy protection, the identity of the person can still be easily inferred from other cues even if the entire body of the person is blurred. Example revealing cues include characteristic clothing, personal items and simi-

* Corresponding author,
E-mail addresses: karla.brkic@fer.hr (K. Brkić), tomislav.hrkac@fer.hr (T. Hrkać), zoran.kalafatic@fer.hr (Z. Kalafatić).

[1] Directive 95/46/EC, http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:31995L0046

http://dx.doi.org/10.1016/j.eswa.2017.05.067
0957-4174/

(a) (b)

Fig. 1. Shortcomings of de-identification by blurring. The person is blurred, but there are still many identifying features remaining: clothing color and texture, body shape, personal bag etc.

lar, as illustrated in Fig. 1[2]. Additionally, it has been shown that blurring itself is easily thwarted by a re-identification attack called parrot recognition (Newton, Sweeney, & Malin, 2005), enabling an attacker with access to another image of the same person to automatically determine the identity of the person.

To address the shortcomings of simpler forms of de-identification and advance the state of the art, we introduce a computer vision-based de-identification pipeline that enables automated pixel-precise segmentation of humans in videos and effective concealment of their identities. In contrast to blurring and similar approaches, our pipeline de-identifies soft and non-biometric features, at the same time preserving the utility of the data. The pipeline consists of three stages: (i) pedestrian detection, (ii) pedestrian segmentation and (iii) de-identification. Assuming a surveillance scenario in which the camera is static and the motion in the scene is mainly due to passing pedestrians, we obtain initial estimates of person locations using a background subtraction algorithm based on mixtures of Gaussians (Zivkovic, 2004). Pixel-precise segmentation of persons is achieved using our improved version of the GrabCut algorithm (Hrkać & Brkić, 2015; Rother, Vladimir, & Blake, 2004). De-identification is performed by transferring the style of another image to the segmented person image through the use of the neural art algorithm (Gatys, Ecker, & Bethge, 2015a) and blending the result with the original image.

2. Related work

The three stages of our pipeline address three research topics that are often studied independently: (i) pedestrian detection, (ii) pedestrian segmentation and (iii) de-identification.

2.1. Pedestrian detection

Pedestrian detection has been a very active topic of research in recent years, resulting in a considerable amount of proposed detectors. Some of the most widespread include the seminal HOG detector based on oriented gradients (Dalal & Triggs, 2005), the detector based on AdaBoost and Haar-like image features (Viola & Jones, 2001) and its extension (Viola, Jones, & Snow, 2005), the detector based on integrating local and global cues via probabilistic top-down segmentation (Leibe, Seemann, & Schiele, 2005), the detector based on pictorial structures (Andriluka, Roth, & Schiele, 2009), the detectors based on integral channel features (Benenson, Omran, Hosang, , & Schiele, 2014; Dollar, Tu, Perona, & Belongie, 2009), and deformable part models (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010). In line with current interest of the computer vision community in deep learning, there are also approaches for pedestrian detection based on convolutional neural networks (Ouyang & Wang, 2012; Ouyang, Zeng, & Wang, 2013; Sermanet, Kavukcuoglu, Chintala, & LeCun, 2013).

Comparative studies of the most popular pedestrian detectors (Benenson, Omran, Hosang, , & Schiele, 2014; Dollar, Tu, Perona, & Belongie, 2009; Dollar, Wojek, Schiele, & Perona, 2012; García-Martín & Martínez, 2015; Zhang, Benenson, Omran, Hosang, & Schiele, 2016) indicate that there are still many open challenges in achieving reliable pedestrian detection. In (Dollar, Wojek, Schiele, & Perona, 2012), the problems that modern pedestrian detectors face are identified by benchmarking a total of sixteen detectors on six datasets. It is shown that the performance of all the considered detectors leaves much to be desired even in ideal conditions, e.g. 20-30% of all pedestrians are missed when dealing with large scale pedestrians in the scene, no occlusions, and requiring a maximum of one false alarm per 10 images. When introducing unfavorable conditions (small scale pedestrians, occlusions) performance degrades even further. In (Benenson, Omran, Hosang, , & Schiele, 2014), more than 40 detectors that have reported results on the Caltech pedestrian detection benchmark are studied. It is qualitatively analyzed which components of the detectors contribute most to the achieved detection rates. The analysis indicates that better performance is driven by using better features, additional data, and context information. The best performing detector combines several techniques within the integral channel features framework.

In order to protect the privacy of persons in video sequences, we need to ensure that each individual is correctly detected so their identifying features can be obfuscated. Given that the output of static image person detectors alone has been shown to be somewhat unreliable, we utilize the assumption that our camera is static and that our target application is surveillance, meaning that in a simplified scenario it can be expected that the majority of motion in the scene is due to pedestrians. Using the assumption of pedestrian motion, we can obtain rough estimates of pedestrian locations using background subtraction. Even if there are other objects moving in the scene generating false positive detections, erring on the side of false positives is not a major concern in our application, as de-identifying non-persons does no harm, while failing to detect and de-identify persons could jeopardize their privacy.

Background subtraction is a video-based computer vision algorithm that enables labeling moving (foreground) pixels and static (background) pixels in each frame. The background is represented using a model, and the foreground in each frame is determined by subtracting the background model from the frame. There are many different background subtraction algorithms that vary in the manner in which the background model is constructed (Brutzer, Hoferlin, & Heidemann, 2011; Cheung & Kamath, 2004; Herrero & Bescós, 2009). A comparative overview of several background subtraction algorithms for detecting pedestrians and vehicles in urban scenes can be found in (Cheung & Kamath, 2004). Considered algorithms include simple frame differencing, median filtering, linear predictive filtering, non-parametric estimate of the pixel density function, approximated median filter, Kalman filter and mixtures of Gaussians. Experimental evidence in the overview suggests that mixtures of Gaussians (Stauffer & Grimson, 1999; Wren, Azarbayejani, Darrell, & Pentland, 1997; Zivkovic, 2004) produce

[2] Image by Juanjo Zanabria Masaveu, licensed under CC BY 2.0.

the best results. Mixtures of Gaussians are also found to be one of the top-performing methods in a comparative study (Herrero & Bescós, 2009), alongside χ^2 modeling and simple median filtering. Given these findings, we use mixtures of Gaussians as our background subtraction algorithm to obtain an initial estimate of the locations of persons. Specifically, we employ adaptive Gaussian mixture model-based background subtraction (Zivkovic, 2004).

While in this work we assume a simplified surveillance scenario and consider background subtraction a sufficiently good estimate of candidate pedestrian locations, in general surveillance scenarios this assumption does not always hold, as there can be other moving objects in the scene. If this is the case, we envision several strategies for improving the detection results: (i) background subtraction outputs can be pruned using outputs of a pedestrian detector (see a comprehensive overview of pedestrian detectors for surveillance (García-Martín & Martínez, 2015)) , (ii) prior knowledge of certain areas that contain moving objects, but not people, could be incorporated in the system (e.g. road regions), (iii) additionally, the detector confidence maps could be used to further separate people and background, as in (García-Martín, Cavallaro, Martínez, & Martínez, 2012). The goal should be to utilize as many cues as possible to improve detection. Finally, if background subtraction itself is completely unreliable in the target application, one should consider switching to more complex motion-based pedestrian detectors (García-Martín & Martínez, 2015) or using a more sophisticated multi-cue detection system, as e.g. in (García-Martín & Martínez, 2012), where a complex detection system is proposed that integrates appearance, motion and tracking information.

2.2. Pedestrian segmentation

Having a rough estimate of pedestrian locations obtained by background subtraction, we apply a segmentation algorithm to obtain well separated pedestrian silhouettes. Our segmentation algorithm is based on the GrabCut algorithm (Rother, Vladimir, & Blake, 2004) for segmenting objects in static images. The original GrabCut algorithm is semi-automatic in the sense that the user is required to draw a rectangle around an object. The area outside the rectangle is considered to definitely belong to background, while the area inside the rectangle is considered to be an approximation of the foreground. Alternatively, the user can specify areas belonging to foreground and background by selecting foreground and background regions using a brush in a graphic editor. The algorithm maintains the models of foreground and background based on Gaussian mixtures. The segmentation task is formulated as an energy minimization problem and solved by an iterative graph cut optimization technique, as proposed in (Boykov & Jolly, 2001). There are two notable weaknesses of the original GrabCut algorithm: first, it is very supervised and only semi-automatic, and second, poor segmentation often occurs in certain cases that are common in real world sequences. For example, poor segmentation can occur when parts of the object share characteristics with parts of the background, when high contrast color changes are present in the background near the object or inside the object, or when the object is concave. We propose several improvements to the original GrabCut algorithm that are specifically designed to overcome its limitations. These improvements in part rely on the output of background subtraction, which we use for segmentation initialization and as prior in other parts of the algorithm.

Several researchers have also considered combining background subtraction with GrabCut. As noted in (Sun, Tang, & Shum, 2006), straightforward use of the result of background subtraction as a mask for GrabCut often gives unsatisfactory results if the static background contains high-contrast elements. To address this problem, they propose an adaptive background contrast attenuation method. The method assumes that what is background is known from background subtraction, and then attenuates the contrast in the background, simultaneously preserving the contrast at the foreground/background boundary. In (Poullot & Satoh, 2014), an algorithm called VabCut is proposed for video foreground object segmentation in videos taken using a moving camera. VabCut extends the RGB color domain with a motion layer M, calculated after RANSAC-based frame alignment. Bounding box and a larger super bounding box around the moving object are calculated and only the area between these two bounding boxes is used for background modeling, in order to avoid visual similarities between foreground and background in case of large backgrounds. Additionally, the numbers of Gaussians in the Gaussian mixtures for foreground and background models are independently optimized. In (Hernandez-Vela, Reyes, Ponce, & Escalera, 2012), tracking and segmentation are combined and a fully automatic spatio-temporal GrabCut human segmentation method is proposed. GrabCut initialization is performed by combining several detectors: HOG pedestrian detector, a face detector, and a skin color model. Segmentation results in concave regions (typical in images of humans) are improved by refining the background mask through adding to it the pixels that have greater probability of belonging to the background. This information is calculated based on foreground and background color models. Temporal component is utilized by favoring segmentations that are close to the results obtained in the previous frame.

Through the combination of background subtraction and our improved GrabCut algorithm we are able to determine which pixels belong to individual persons in each considered video frame. Focusing on the classification of individual pixels, rather than finding bounding boxes around the persons, ensures that the outline of each person can be retained in the de-identified sequence. Retaining the outline adds to the understandability of the scene, while simultaneously ensuring maximum protection of privacy through de-identification of the person pixels denoted by the outline.

2.3. De-identification

The interest in computer vision-based de-identification has been growing in recent years, with a number of methods being proposed (Gross, Sweeney, Cohn, De la Torre, & Baker, 2009; Padilla-López, Chaaraoui, & Florez-Revuelta, 2015; Ribarić, Ariyaeeinia, & Pavešić, 2016). While primary focus still seems to be on de-identifying faces only (Gross, Sweeney, Cohn, De la Torre, & Baker, 2009), there are methods devoted to full body de-identification (Agrawal & Narayanan, 2011; Park & Trivedi, 2005), as well as works on soft and non-biometric features detection and recognition (Han & Jain, 2013; Heflin, Scheirer, & Boult, 2012; Kim, Parra, Yue, Li, & Delp, 2015; Reid, Samangooei, Chen, Nixon, & Ross, 2013). Simple approaches to face de-identification include pixelization, blurring and applying various image distortions (Gross, Sweeney, Cohn, De la Torre, & Baker, 2009). As noted previously, although seemingly effective, these kinds of approaches have been shown to be vulnerable to re-identification attacks. In (Newton, Sweeney, & Malin, 2005), it is shown that a classifier trained on blurred or otherwise transformed images of a person can easily recognize that person in other, previously unseen images. As a solution, the *k*-same algorithm is introduced (see also the extensions of the algorithm in (Gross, Airoldi, Malin, & Sweeney, 2006a; Gross, Sweeney, de la Torre, & Baker, 2006b)). The *k*-same algorithm clusters face images and computes the average of each cluster. All of the face images belonging to one cluster are replaced with the average face. The idea of replacing the face with something else is utilized in other works as well. In (Lin, Wang, Lin, & Tang, 2012) face swapping is performed by building 3D head models for frontal faces. In (Bitouk, Kumar, Dhillon, Belhumeur, & Nayar, 2008), a sys-

tem for face replacement that relies on selecting a similar face from a database of face images is introduced.

In this work, we introduce a novel de-identification approach that utilizes the neural art algorithm (Gatys, Ecker, & Bethge, 2015a) to de-identify persons in videos. We have previously proposed a similar approach intended for faces only (Brkić, Hrkać, Sikirić, & Kalafatić, 2016). The algorithm uses the responses of a deep neural network to transfer the style of one image to another. In the original work (Gatys, Ecker, & Bethge, 2015a), the transfer is done from an artwork to a target photograph, i.e. the algorithm renders the photograph in the style of the artwork. This means that the content of the photograph, i.e. global structure and arrangement, remains preserved, while the style, i.e. colors and local structures, is obtained from the artwork. We apply the neural art algorithm on the segmented pedestrian images using a database containing a wide variety of style images, including realistic photographs, artworks and synthetic renderings. Through altering the style of the segmented images and preserving the content, we obtain images that still distinctly represent humans, but with changed appearance features, in effect de-identifying them.

Our work advances the state of the art in computer vision-based de-identification through proposing a complete pipeline for de-identification of surveillance videos that de-identifies both hard biometric features (e.g. the face) and soft and non-biometric features (e.g. hair color and clothing). Individual elements of the pipeline also present more fundamental contributions in terms of segmentations of objects in video and human appearance altering through novel applications of the neural art algorithm. In the following sections, we give a detailed overview of the stages of our pipeline.

3. Pedestrian detection and segmentation

The first two stages of our pipeline are (i) pedestrian detection and (ii) pedestrian segmentation. The detection step is based on background subtraction and provides rough estimates of candidate pedestrian locations. These estimates are then precisely segmented using our improved GrabCut algorithm. The algorithm is the result of our previous work (Hrkać & Brkić, 2015). In this section we describe the algorithm in detail for the sake of completeness.

3.1. Obtaining initial background subtraction estimates

When an image enters our de-identification pipeline, we start by obtaining an initial foreground – background estimation using background subtraction based on Gaussian mixture models (GMMs) (Zivkovic, 2004). In this algorithm, the distributions of values of each pixel over time are modeled with a mixture of weighted Gaussian distributions. As a new frame arrives, the weights of the Gaussians are updated. The influence of older frames diminishes over time. In order to estimate foreground and background, the algorithm assumes that the highest weights in the Gaussians for each pixel will be assigned to background, given a sufficiently wide time window. In other words, when observing each pixel through time, the color that most commonly appears is likely to belong to the background. This assumption holds in a surveillance scenario where it is reasonable to expect that the number of frames in which the background is occluded is smaller than the number of frames in which the background is visible.

The background subtraction algorithm is a probabilistic foreground/background mask, as illustrated in Fig. 2 (b) (the original image is shown in Fig. 2 (a)). As can be seen on this example, the resulting foreground pixels often do not cover all pedestrian pixels. Some pedestrian pixels are marked as background, and there is some noise in the output. Our goal in this stage of the pipeline is to label all pedestrian pixels as foreground. We binarize the output foreground/background mask and use it as input to an algorithm for contour extraction and filling, resulting in an improved mask, as illustrated in Fig. 2 (c) . To remove small and unconnected regions and noise, we follow this step with applying the morphological operation of closing (dilation followed by erosion), as illustrated in Fig. 2 (d). Through these steps, we obtain a foreground mask that roughly corresponds to the desired pedestrian segmentation, but is too coarse to be used as a final result. The resulting coarse foreground mask is used as an input to our improved GrabCut algorithm that smooths the contours and provides precise pedestrian segmentations.

As mentioned previously, this approach to pedestrian detection relies on motion as a detection cue, so non-pedestrian objects moving in the scene will trigger false positive detections. However, assuming that the number of these false positives is sufficiently small not to jeopardize the naturalness of the scene, de-identifying these false positives does not present a problem. In applications where there is no motion, or where there are many non-human moving objects, more complex detection strategies should be used (see the discussion in Section 2.1).

3.2. The improved GrabCut algorithm

The second stage of our pipeline is pedestrian segmentation achieved using an improved version of the GrabCut segmentation algorithm. We apply the foreground mask obtained in the detection stage as a foreground prior for the segmentation.

3.2.1. An overview of the original GrabCut algorithm

The original GrabCut algorithm (Rother, Vladimir, & Blake, 2004) is intended for user-supervised segmentation of objects in images. Example scenarios necessitating the use of user-supervised segmentation include common image editing tasks, medical segmentation etc. By either drawing a rectangle around the object or using a brush to highlight parts of the object and/or background, the user implicitly provides a prior estimate on two categories of pixels: sure background (areas outside the rectangle or areas brushed as background) and/or sure foreground (areas inside the rectangle or areas brushed as foreground). Formally, we represent the color image I as an array $z = (z_1, \ldots, z_N)$ of N pixels in RGB space, where z_i is a triplet of color values, $z_i = (R_i, G_i, B_i)$. Each pixel z_i is assigned a label α_i, indicating whether it belongs to background ($\alpha_i = 0$) or foreground ($\alpha_i = 1$). Thus, the segmentation of the image is defined by an array $\alpha = (\alpha_1, \ldots, \alpha_N)$, $\alpha_i \in \{0, 1\}$. Internally, the algorithm works with a trimap T over the image, consisting of three regions: T_B, T_F and T_U that specify pixels belonging to sure background, sure foreground, and uncertain pixels, respectively. The initial values of T_B and T_U are set according to user input, while T_F is set to $\emptyset$. In other words, the foreground marked by the user is treated as uncertain pixels, while the background is treated as sure background (a vice versa scheme could also be applied). The algorithm then iteratively labels uncertain pixels as either foreground or background. The value α_i is initialized to 0 for pixels in T_B and to 1 for pixels in $T_F \cup T_U$.

The algorithm keeps two full covariance Gaussian mixture models (GMMs) of K components, one GMM for foreground and one GMM for background. Each of the models is parametrized as:

$$\theta = \{\pi(\alpha, j), \mu(\alpha, j), \Sigma(\alpha, j), \alpha \in \{0, 1\}, j = 1, \ldots, K\}, \quad (1)$$

where the j-th component of the model is defined by its weight π, its mean μ and its covariance matrix Σ. The algorithm uses an array $k = \{k_1, \ldots, k_n\}$, where $k_i \in \{1, \ldots, K\}$ indicates the component of the background or foreground GMM (according to α_i) the pixel z_i belongs to. The segmentation task is formulated as an energy minimization problem, where low energy indicates a good

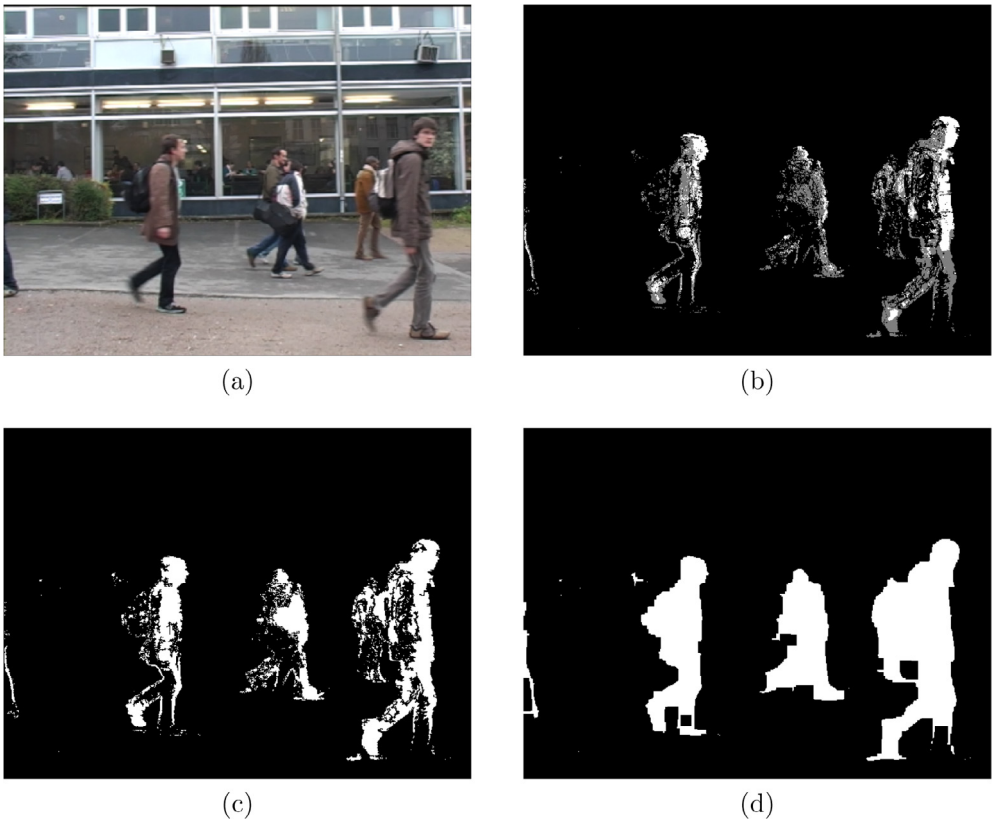

Fig. 2. Background subtraction in the first stage of our pipeline: (a) the original frame, (b) the output of the background subtraction algorithm (levels of gray denote the probability that the pixel is foreground), (c) the extracted and filled contours, (d) the output after morphological closing.

segmentation. The energy is a sum of two components: the so-called data term and the so-called smoothness term:

$$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z), \tag{2}$$

The data term U enforces the consistency of the segmentation with the observed foreground and background models, while the smoothness term V enforces the solidity of the object in terms of color similarity. Specifically, the data term is defined as:

$$U(\alpha, k, \theta, z) = \sum_i (-\log(\pi(\alpha_i, k_i) p(z_i|\alpha_i, k_i, \theta)). \tag{3}$$

The data term takes on small values when the current segmentation results in pixel assignments conforming to prior models of foreground and background, and its value increases as the segmentation diverges from the foreground and background priors.

The smoothness term is defined as:

$$V(\alpha, z) = \gamma \sum_{\{m,n\} \in C} [\alpha_n \neq \alpha_m] \exp(-\beta ||z_m - z_n||^2), \tag{4}$$

where C is a set of pairs of neighbouring pixels (8-way connectivity is used), $[\alpha_n \neq \alpha_m]$ is the indicator function that takes values 0 or 1 according to the truth value of the condition, and β is a parameter that weights the color contrast. The authors of the GrabCut algorithm suggest setting $\beta = (2\langle ||z_m - z_n||^2 \rangle)^{-1}$, where $\langle \cdot \rangle$ is the expectation operator. The expression $\exp(-\beta ||z_m - z_n||^2)$ measures the contrast between the neighbouring pixels, taking on low values if the contrast is high and vice versa. The factor $[\alpha_n \neq \alpha_m]$ ensures that the smoothness term captures the contrast information only along the segmentation boundary. Defining the smoothness term in this way ensures that segmentations where adjacent pixels of similar colors are labeled differently are penalized.

Minimizing the energy function E is performed using the iterated graph cut algorithm (Boykov & Jolly, 2001). Each iteration results in a more precise segmentation, improving the underlying Gaussian mixture models. The iterations can either be repeated until convergence or for a fixed number of times.

3.3. GrabCut limitations and our improvements

As shown previously, rough estimates of pedestrian locations can be obtained via background subtraction and morphological operations, assuming a static camera where the motion is mainly due to passing pedestrians. We propose to use these estimates to automatically initialize GrabCut without the need for human input. For each blob obtained by background subtraction and morphological operations, the idea is to run GrabCut initialized with the blob region as uncertain foreground and every other pixel of the image as sure background. However, a number of problems with the original algorithm stand out when applied in this scenario (Hrkać & Brkić, 2015; Sun, Tang, & Shum, 2006). Namely, GrabCut prefers segmenting uniform colors, while it is quite common for humans to wear a shirt and pants in contrasting colors, which typically results in only a part of the person being segmented. Also, the color of human clothing can often match the color of the background (e.g. a person in a white shirt walking near a white wall), resulting in unwanted segmentation of the background. Finally, the shape of the human silhouette is concave, while GrabCut prefers smooth convex boundaries.

Our improvements are designed to emphasize the importance of the initial silhouette estimate obtained by background subtraction, hence implicitly enforcing motion as an important segmentation cue. First, in building foreground and background models, we weight the influence of each pixel according to its distance from the blob boundary. Second, we modify the probabilities of a pixel belonging to foreground or background in accordance with

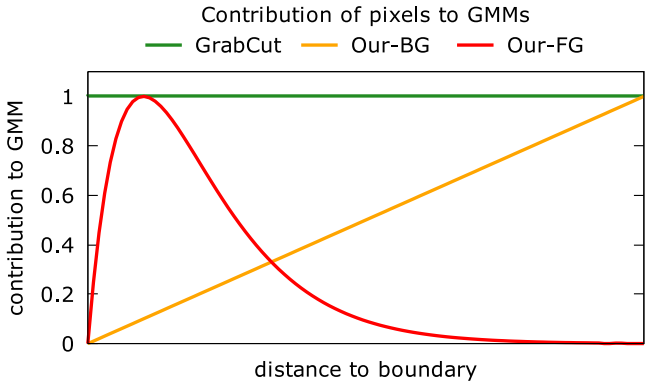

Fig. 3. The weighing of pixels used to construct foreground and background GMMs in our algorithm, as opposed to the original GrabCut.

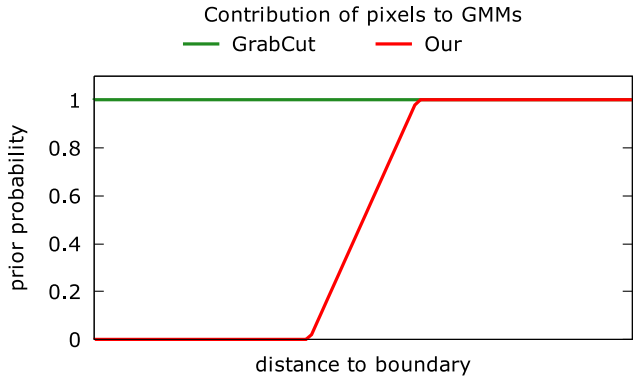

Fig. 4. The prior probability weights for a pixel belonging to foreground or background in our algorithm, as opposed to the original GrabCut algorithm.

the output of background subtraction, taking into account its estimated reliability. Third, we discourage segmentation boundaries that are far away from the initial silhouette estimate, to prevent partial segmentations due to e.g. contrasting garments. We now review each of the improvements in detail.

3.3.1. Improvements to GMM construction

When building foreground and background Gaussian mixture models, we introduce weighing the influence of each pixel depending on where the pixel is in relation with the background subtraction blob. We assume that background subtraction provides a strong prior on approximate object shape, so the pixels that are important for segmentation lie close to the blob boundary. However, based on our experimental observation that pixels lying *exactly on the boundary* tend to be misclassified due to imprecisions introduced by noise and morphological operations, we simultaneously downweigh the influence of pixels very close to the boundary.

The weight factor $w_i(\alpha_i, z_i)$ of a pixel z_i with a preliminary background subtraction-based classification $\alpha_i \in \{0, 1\}$ (background or foreground) is:

$$w_i(\alpha_i, z_i) = \begin{cases} \kappa_F d_{\min}(i) & \text{for } \alpha_i = 1 \\ \kappa_G d_{\min}(i) \exp(-\tau d_{\min}(i)) & \text{for } \alpha_i = 0 \end{cases} \quad (5)$$

where $d_{\min}(i)$ is the distance of the pixel from the prior object boundary, τ is an empirically determined constant, and κ_F and κ_G are normalizing constants used to ensure that $\forall i\, w_i \in [0, 1]$. As can be seen from Eq. 5, the weighting scheme differs for foreground and background pixels, as illustrated in Fig. 3. This scheme is based on our empirical observations of background subtraction outputs, indicating low relevance of pixels on the boundary itself, higher relevance of background pixels with small distances to the boundary with an exponential drop (the pixels close to the boundary are the most important to delineate background and foreground), and linearly increasing relevance of foreground pixels that are further from the object boundary (pixels closer to the center of the blob are more likely to be correctly classified as foreground). After weighting is applied, the energy function is formulated according to the original GrabCut setup (Eq. 2).

3.3.2. Improvements to the data term

In the original GrabCut, the data term U indicates the consistency of the pixel with active background and foreground models. In our improved data term, we also take into account the initial classification of the pixel obtained by background subtraction, weighting the likelihood that the pixel had initially been correctly classified. The reasoning is similar as in the previous section; we introduce a parameter that measures the probability of correct initial classification based on the distance of the pixel from the background subtraction blob boundary. Expecting that the pixels close to the blob boundary will be misclassified, we multiply the original probability of a pixel belonging to the opposite category by a value that decreases with the distance of the pixel from the boundary. Our modified data term is:

$$U(\alpha, k, \theta, z) = \sum_i -\log(P_i(\alpha_i, z_i)\pi(\alpha_i, k_i)\, p(z_i|\alpha_i, k_i, \theta)), \quad (6)$$

where $P_i(\alpha_i, z_i)$ represents the prior probability of each pixel belonging to foreground or background (according to α_i), based on the background subtraction blob. The value $P(\alpha_i, z_i)$ is calculated as:

$$P_i(\alpha_i, z_i) = \begin{cases} 1 \text{ for } d_{\min}(i) > D(\alpha_i) \\ 0 \text{ for } d_{\min}(i) < D(1-\alpha_i) \\ 1 - d_{\min}(i)/D_{\max}(1-\alpha_i) \text{ otherwise} \end{cases}. \quad (7)$$

The factor $D(\alpha_i)$, the so-called distance threshold, is used to regulate how far the pixel should be from the background subtraction blob boundary to consider its preliminary classification as foreground or background as correct. We determine this factor empirically. The value $D_{\max}(\alpha_i)$ is the maximum distance of all foreground or background pixels from the preliminary object boundary. Eq. 7 ensures that for pixels far away from the blob boundary the initial background subtraction-based classification is strongly favored. For pixels near the boundary, the original GrabCut probability of belonging to the opposite category is multiplied by a factor that decreases with the distance of the pixel to the boundary. An illustration of prior probabilities in our algorithm compared with the original GrabCut is shown in Fig. 4.

3.3.3. Improvements to the smoothness term

In the original GrabCut, the smoothness term V ensures that the segmentation boundary is not placed at locations where neighboring pixels share the same color, strongly favoring placing the boundary between contrasting regions. In our application, contrasting regions corresponding to differently clothed garments the person is wearing are common, as are situations when one of the garments of the person matches the color of the background. Additionally, long and concave blob boundaries, as are the boundaries of pedestrians, strongly contribute to the total energy. When keeping the original smoothness term, the algorithm prefers short boundaries, resulting in cutting through the object and/or erroneously segmenting parts of the background as foreground. Therefore, we propose a modified smoothness term that discourages placing boundaries far away from the initial background subtrac-

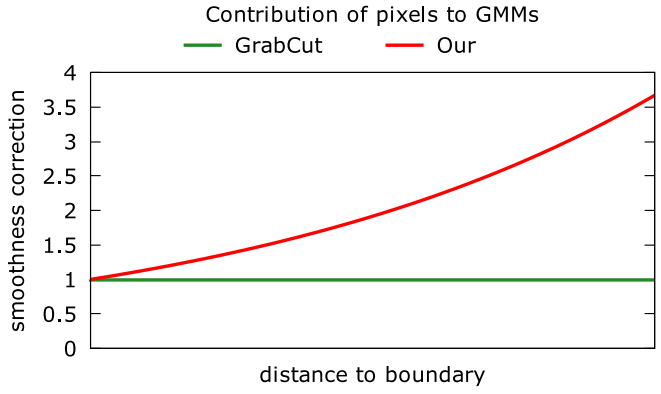

Fig. 5. Smoothness correction in our algorithm, as opposed to the original GrabCut algorithm.

tion blob boundary and through the background:

$$V(\alpha, z) = \gamma \sum_{\{m,n\}\in C} [\alpha_n \neq \alpha_m] \exp(-\beta(||z_m - z_n||^2 + \lambda(d_{\min}(i) - \delta_0))). \tag{8}$$

As can be seen from Eq. 8, the original smoothness term (see Eq. 4) is multiplied by another exponential, $\exp(\beta\lambda(d_{\min}(i) - \delta_0))$, where λ and δ_0 are empirically determined constants. The constant δ_0 ensures that the penalization is not high for pixels near the background subtraction blob boundary, while λ is a normalizing factor intended to scale the influence of the additional exponential to the influence of the original smoothness term. The illustration of the effects of this change to the smoothness term compared to the original GrabCut algorithm is shown in Fig. 5.

3.4. An example of performance

Our improved GrabCut combined with background subtraction provides high quality person silhouettes with precise boundary segmentations and very little noise, as shown in an illustrative example in Fig. 6. We see that the original GrabCut has problems with cutting through the pedestrians due to contrasting garments worn, while our improved version correctly segments all pedestrians. Exact knowledge of which pixels belong to a person is a prerequisite for adequate de-identification, that is in our pipeline achieved through neural art.

4. Neural art-based de-identification

The main idea of the de-identification step in our pipeline is to utilize the neural art algorithm to obfuscate the appearance of the segmented pedestrian. We used similar idea in our previous work (Brkić, Hrkać, Sikirić, & Kalafatić, 2016) where neural art was used for de-identification of face images, while here we propose an approach to de-identify the appearance of whole pedestrian silhouettes.

In general, the neural art algorithm is intended for transferring the style of one image to the content of another, utilizing the responses of a deep neural network. An illustration is shown in Fig. 7. In this work, we choose style images that result in obfuscating the appearance of persons, including obfuscating faces, clothing style, clothing color, skin color etc.

The neural art algorithm maintains two representations: a representation of content of the target image, and a representation of style of the image that is the source for style transfer. The result image is obtained by a joint minimization of the distance of a white noise image from the style and the content images. We now review the algorithm in detail.

4.1. Content representation

When looking at a convolutional neural network, the position of a network layer in the processing hierarchy determines the amount of contextual information that layer encodes. Earlier layers usually specialize for raw pixel values and low-level image features, while layers further along the processing hierarchy encode higher level concepts, such as objects and their relations. The neural art algorithm utilizes this fact to built its content representation. Assuming a layer of n_l feature maps, with the total dimension of each feature map (its width times its height) equal to m_l, we define the $n_l \times m_l$ matrix F^l as a matrix of network responses in layer l. The element $F^l(i, j)$ is the response of the i-th feature at position j. The encoded content at a given layer is reconstructed by starting with an initial white noise image and performing gradient descent optimization until an image that matches the feature responses of the original image is obtained. The loss function is defined as:

$$\mathcal{L}_{\text{content}}(I_C, I_G, l) = \frac{1}{2}\sum_{i,j}(F^l_G(i, j) - F^l_C(i, j))^2, \tag{9}$$

where I_C is the image containing the modeled content with the matrix of responses F^l_C, while I_G is the generated image with the matrix of responses F^l_G.

The gradient is computed using standard error backpropagation, using the derivative of the loss:

$$\frac{\partial \mathcal{L}_{\text{content}}}{\partial F^l_G(i, j)} = \begin{cases} F^l_G(i, j) - F^l_C(i, j) & \text{if } F^l_G(i, j) > 0 \\ 0 & \text{if } F^l_G(i, j) \leq 0. \end{cases} \tag{10}$$

The optimization starts with a random image I_G, and the image is updated until the difference between feature responses is minimized.

4.2. Style representation

The style representation of an image in the neural art algorithm at a given network layer is obtained by computing the correlations between feature responses at that layer (Gatys, Ecker, & Bethge, 2015b). The correlations are defined by a Gram matrix G^l with n_l rows and columns, where $G^l(i, j)$ is the inner product of flattened vectors of feature responses i and j. To find a texture image that matches the style of the input image, a random image is initialized and gradient descent optimization is performed with respect to the style representation. The loss function $\mathcal{L}_{\text{style}}(I_S, I_G)$ is the sum of losses E_l across all network layers:

$$\mathcal{L}_{\text{style}}(I_S, I_G) = \sum_l w_l E_l, \tag{11}$$

where I_S is the style image, I_G is the generated image and w_l are the contributing weights of each layer. The loss of an individual layer E_l is defined as:

$$E_l = \frac{1}{4n_l^2 m_l^2}\sum_{i,j}(G^l_G(i, j) - G^l_S(i, j))^2, \tag{12}$$

where G^l_S is the Gram matrix of the style image and G^l_G is the Gram matrix of the generated image, n_l is the number of feature maps, and m_l is the total dimension of each feature map.

The derivatives of E_l with respect to activations in layer l are:

$$\frac{\partial E_l}{\partial F^l(i, j)} = \begin{cases} \frac{(F^l)^T(G^l_G(i,j) - G^l_S(i,j))}{n_l^2 m_l^2} & \text{if } F^l(i, j) > 0 \\ 0 & \text{if } F^l(i, j) \leq 0. \end{cases} \tag{13}$$

4.3. Generating the mixed images

To generate the image that mixes the content of one image with the style of another, the neural art algorithm initializes a white

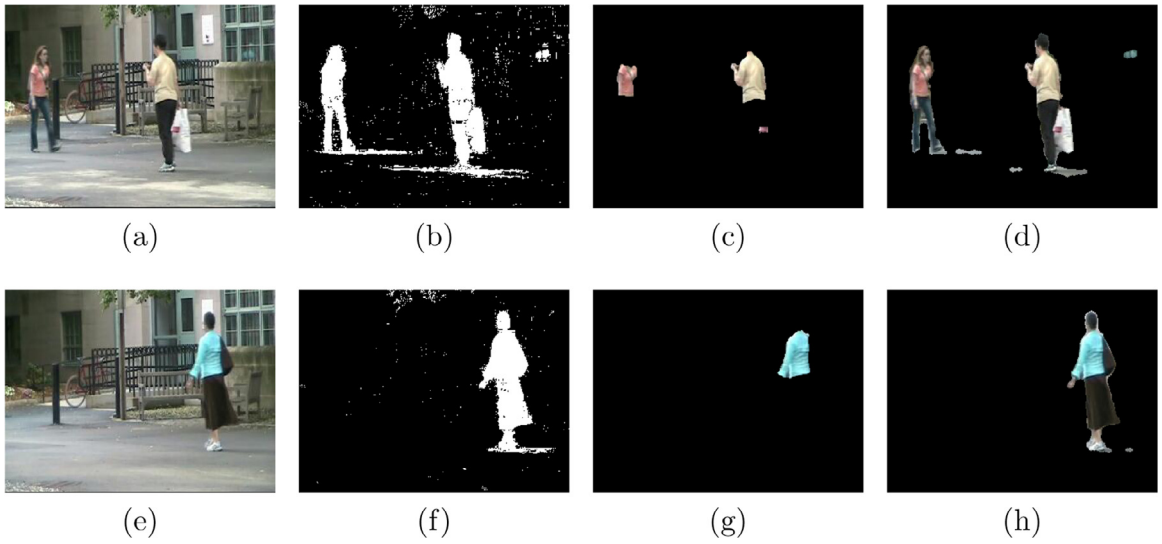

Fig. 6. Output comparisons for two frames from the sequence "backdoor" from the CDnet 2014 Pedestrian Detection dataset (Wang et al., 2014): (a,e) original frames, (b,f) raw background subtraction output, (c,g) GrabCut, (d,h) our method.

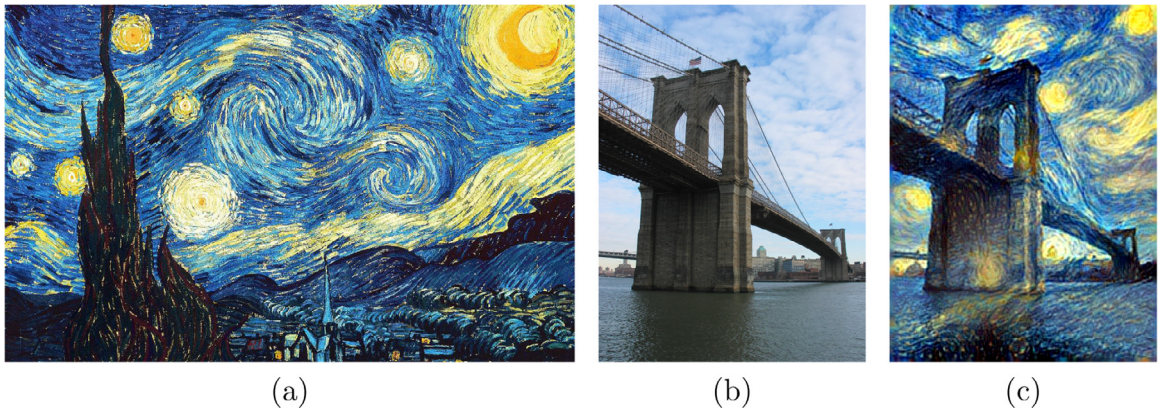

Fig. 7. Example output of the neural art algorithm (Gatys, Ecker, & Bethge, 2015a). The style of the artwork (a) is mixed with the content of the image (b) to obtain the output image (c).

noise image and jointly minimizes the content loss in one layer and the style loss across multiple levels. The loss function is:

$$\mathcal{L}_{\text{total}}(I_C, I_S, I_G) = \alpha \mathcal{L}_{\text{content}}(I_C, I_G) + \beta \mathcal{L}_{\text{style}}(I_S, I_G). \quad (14)$$

The end result of this optimization is the image I_G that is a mix of content from the image I_C and style from the image I_S.

4.4. De-identification and its reversibility

In order to de-identify segmented persons in our pipeline, we alter their style using the neural art algorithm. We limit the application of the algorithm to the segmented persons only, so the background remains natural and untampered with. In practice, this is achieved by applying the neural art algorithm on the entire input image, cutting the resulting image according to the segmented persons' silhouettes and blending the result back with the input image.

In terms of source style images, we use a database of richly textured images with lots of details. Any kind of photograph, artwork or synthetic image could be used, lending more or less naturalness and credibility to the final de-identified image. In order to de-identify a person, we randomly select a style image from the database, apply the neural algorithm and integrate the result into the original image.

One example of de-identification with neural art is shown in Fig. 8. The person from the content image (a) is segmented, the style from the style image (b) is applied on the entire content image, and the result is cut out and blended back with the original image (c). It can be seen that the method has many desirable properties of good de-identification: it is recognizable what the person is doing, but the appearance of the person has been altered, with changed clothing, shoes, hairstyle, body shape, and an added hint of a personal bag that was not there before. It has been made difficult to recognize the person from secondary cues, i.e. soft and non-biometric identifiers.

In most de-identification applications, the ability to be able to securely store and, if need be, reverse the de-identifying transformation is one of the key requirements. In our pipeline, we envision that the reversibility could be achieved by either (i) securely encrypting and separately storing the original image, or (ii) steganographically encoding the original image (either encrypted or unencrypted, depending on the level of protection needed) in the de-identified image itself (see e.g. our work (Blažević, Brkić, & Hrkać, 2015)). In applications that involve crowded scenes, separate encryption of the original image is a better solution, as larger volumes of steganographically-encoded data include progressive deterioration of the carrier image quality.

Fig. 8. Neural art-based de-identification. The pedestrian from the image (a) is segmented, combined with the style image (b) and the result is pasted into original image (c).

5. Experiments

The experimental evaluation of our pipeline consists of two parts: (i) the evaluation of pedestrian detection and segmentation, and (ii) the evaluation of de-identification.

5.1. Pedestrian detection and segmentation

In order to experimentally validate the performance of our method in terms of pedestrian detection and segmentation, we employ sequences from the CDnet 2014 Pedestrian Detection dataset (Wang et al., 2014), containing ten videos with a total of 26,248 frames. Example frames from each of the ten videos are shown in Fig. 9. We compare the performance of our improved GrabCut algorithm to a reference implementation of the original GrabCut from the OpenCV library (Bradski, 2000). Both algorithms are initialized using the same background subtraction blobs obtained after morphological postprocessing. Pipeline parameters are initialized as follows: morphological opening of 5 pixels, morphological closing of 9 pixels, $\tau = 10$. $D(\alpha_i) = 0.2 \cdot D_{\max}(\alpha_i)$, $\lambda = 20000$, $\delta_0 = 8$. The values were determined empirically.

We adhere to the evaluation methodology for the CDnet 2014 dataset proposed in (Wang et al., 2014), evaluating the performance on a per-pixel basis. We measure average precisions, recalls and F1 measures over all frames for each sequence in the dataset. The results are shown in Table 1. In terms of F1 measure (the harmonic mean of precision and recall), our method outperforms the reference GrabCut implementation in all the considered sequences. The improvement of the F1 measure is caused by a significantly improved recall, as many more foreground pixels are correctly segmented when our algorithm is applied instead of the original GrabCut implementation. However, our algorithm produces somewhat more false positives classified as foreground, resulting in a relatively small drop of precision.

An example comparing the performance of our algorithm with the original GrabCut is shown in Fig. 6. As can be seen, the original GrabCut performs poorly as there is a significant color difference in individual clothing items worn by each of the pedestrians. On the other hand, our algorithm correctly segments all pedestrians, as our improvements ensure that the background subtraction blob outline is smoothed, while pixel contributions become dependent on the position of the pixels relative to the boundary subtraction blob. In terms of run time performance, our algorithm runs at a speed similar to the reference implementation of GrabCut.

Table 1
Average precisions, recalls and F1 measures for each of the ten sequences in the CDnet 2014 Pedestrian Dataset.

Sequence	GrabCut			Ours		
	AP	AR	F1	AP	AR	F1
backdoor	0.9938	0.3784	0.5481	0.9298	0.9175	0.9236
bus station	0.9770	0.1186	0.2115	0.8569	0.5039	0.6346
cubicle	0.9901	0.3642	0.5325	0.7339	0.7924	0.7620
copy machine	0.9900	0.2266	0.3687	0.5847	0.3988	0.4741
office	0.9943	0.1184	0.2116	0.7625	0.2400	0.3650
pedestrians	1.0000	0.5917	0.7434	1.0000	0.9383	0.9681
PETS 2006	1.0000	0.4171	0.5886	1.0000	0.7169	0.8351
people in shade	0.8890	0.5364	0.6690	0.7555	0.7233	0.7390
skate	1.0000	0.5102	0.6756	1.0000	0.6788	0.8086
sofa	0.9659	0.1651	0.2819	0.6841	0.5449	0.6066

We conclude that the detection and the segmentation stages of our pipeline provide reliable pedestrian detections with precise silhouettes. The number of false positives produced by the segmentation stage is somewhat higher than when the original GrabCut is used, but in the context of de-identification this is not a concern (we do not expect the naturalness of the scene to be severely impacted by de-identifying a small number of pixels not belonging to pedestrians).

5.2. De-identification

Our experimental evaluation of the effectiveness of the proposed de-identification technique consists of two complementary approaches. The first approach focuses on automatically evaluating the effects of de-identification by measuring the performance of detection and classification algorithms on original vs. de-identified images. The second approach systematically studies how humans perceive the de-identified images using a series of questionnaires. Our rationale is that for a de-identification method to be considered successful, it must thwart the identification of subjects both by automated methods and by human observers.

5.2.1. The employed datasets

As described previously, the neural art algorithm that we use for de-identification requires two images as input: a content image that is to be transformed, and a style image whose style is used to transform the content image. In these series of experiments, we use walking sequences from the Human3.6m dataset (Ionescu, Papava, Olaru, & Sminchisescu, 2014) as our content images. The sequences are filmed with a static camera of a relatively high resolution (1000 × 1002 pixels), and depict different subjects walking. The Human3.6m dataset was selected for our de-identification experiments because the high resolution of the images meant that the effects of de-identification on individual features on the face and the body would be more pronounced. In contrast, in the CDNet 2014 dataset that we used for detection and segmentation experiments pedestrians tend to be very small, so it is often impossible to discern their facial features and other details of their appearance, therefore making it hard to closely study the effects of de-identification.

While the content images for de-identification experiments come from the Human3.6m dataset, for style images we need a more versatile dataset containing a variety of styles, local textures and colors. Therefore, we have manually compiled a dataset of 23 diverse images primarily obtained from ImageNet

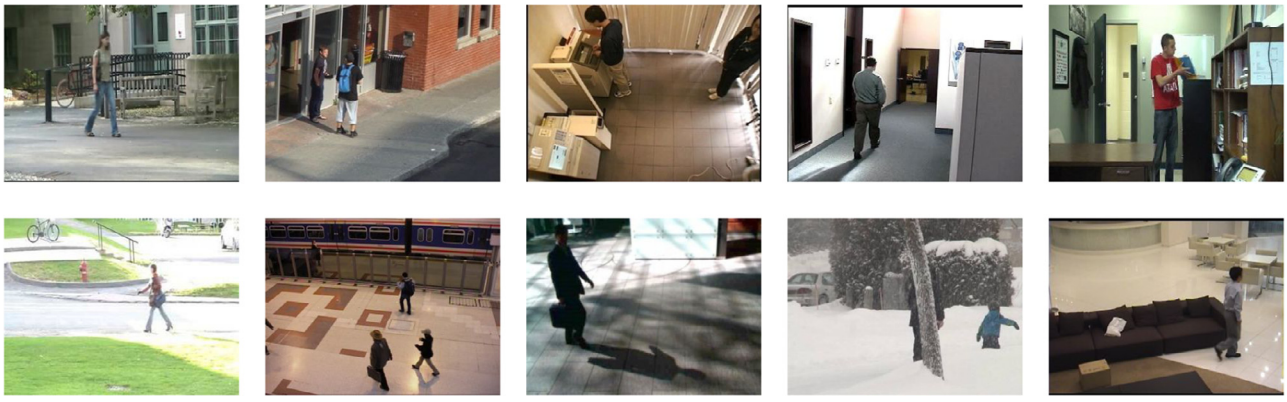

Fig. 9. Example frames from the CDnet 2014 Pedestrian Dataset videos.

Fig. 10. A few style images used for de-identification.

(Russakovsky et al., 2015). The selected style images range from everyday situations depicting people to textures and artwork. A few of the style images can be seen in Fig. 10.

5.2.2. Qualitative evaluation

Some results of our initial, qualitative evaluation of de-identification using the described datasets are shown in Fig. 11. The first row depicts the used style images, the first column the original frames, and combined images are shown in corresponding rows and columns. While the pose and the body shape of the persons remains visible in all de-identified images, we see considerable changes in their appearance. Clothing appears to be colored and textured differently and in some cases replaced by a different type (shirts and pants can become differently colored dresses). Bare skin sometimes gets covered by made up clothes. Facial features tend to get obfuscated or removed. The changes are dependent both on style and content. As can be seen, not every style changes the content image in the same way. In the first column, the clothing of the person in the first row turns blue due to the selection of blue features from the style image, while the clothing of the person in the second row is re-colored using yellowish features and textures from the style image. However, the sneakers of the person are rendered blue.

Depending on the applied style image, the de-identified images could look somewhat unnatural and create a distraction for human observers such as security personnel. However, this holds for all de-identification methods and we do believe that our method offers a better degree of naturalness than some similar works. Furthermore, unlike pixelization, blurring and scrambling, the proposed method keeps a higher degree of data utility. In Fig. 12 we show several examples of alternative de-identification approaches, such as pixelization, blurring and DCT-block scrambling similar to (Dufaux & Ebrahimi, 2008).

Qualitative evaluation suggests that through applying our de-identification pipeline we can obfuscate or remove many biometric, soft biometric and non-biometric identifiers (e.g. face, hair color, birthmarks, tattoos, clothing, skin color etc.) while still retaining information about what is occurring in the scene (in this case, a person is walking). We now verify these observations quantitatively.

5.2.3. Detection performance on de-identified images

To investigate whether the body shapes and the faces of the persons remain automatically detectable after applying our de-identification method, we employ four detectors: (i) a pedestrian detector based on deformable part models (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010), (ii) a pedestrian detector based on integral channel features (Dollar, Tu, Perona, & Belongie, 2009), (iii) a pedestrian detector based on histograms of oriented gradients (HOG) (Dalal & Triggs, 2005) and support vector machines (SVM) (Cortes & Vapnik, 1995), (iv) the Viola-Jones object detector (Viola & Jones, 2001) based on boosted Haar cascade classifiers, trained for detecting either just the face or the whole upper body. We use open source implementations from the CCV and OpenCV libraries (Bradski, 2000).

We build a series of de-identified images using sequences of five subjects from the Human3.6m dataset and all style images from the compiled style dataset, resulting in a total of 1127 de-identified images created from 49 original subject images. We then compute the baseline performance of individual detectors on the original images and compare it to performance on de-identified images. The results are summarized in Table 2.

As can be seen, the detection performance of the DPM detector trained on full body is similar on original and de-identified images (the detection rate on the de-identified images is decreased by 5%). This result is to be expected, given that DPM is a detector based on capturing the shape of the detected object. In our case, the de-identified silhouettes of humans still retain a natural human shape, therefore triggering the detection. Detection rates are also similar for original and de-identified images for ICF, HOG and the Viola-Jones detector trained on upper body. We note that the performance of the ICF detector is in itself low and we attribute this to the fact that we used an open source implementation trained on fully front-facing pedestrians, while in the frames from the Human3.6m dataset pedestrians are filmed from various angles.

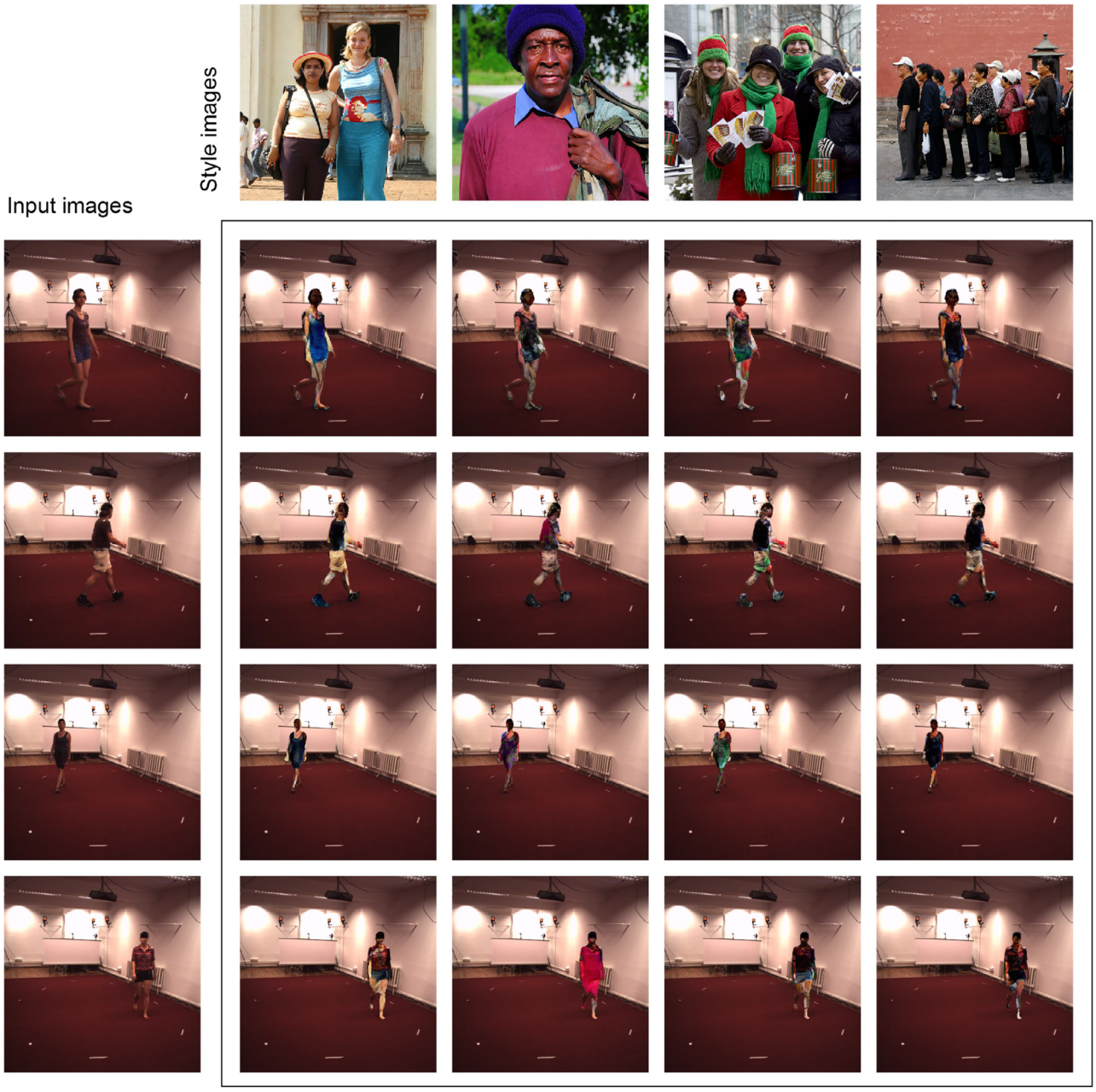

Fig. 11. Applying four different styles (first row) to four different input images (first column). Level of transparency is 0.8.

Face detection using the Viola-Jones detector drops more than threefold on de-identified images, as facial structures are obfuscated through de-identification. We note the initial low performance of the Viola-Jones face detector (accuracy of only 35%) which we attribute to the fact that in the test sequences the subject faces are often very small and poorly discernible.[3]

In summary, experimental results regarding the detection of the persons de-identified by our pipeline indicate that the overall shape of the person remains detectable at a similar rate as in the original images. Should the target application call for lower full body detection rates, these could be achieved by altering the overall shape of the person (e.g. using random distortions on the segmented outline), therefore sacrificing a degree of naturalness in the scene. In this work, our goal is to preserve as much naturalness as possible while obfuscating identifying features. Therefore, while preserving the overall body shape, our method alters facial features so that the face detection rate is more than three times smaller compared to the face detection rate on the original images. Only 11% of faces are correctly detected.

5.2.4. Classification performance on de-identified images

We have shown that our de-identification pipeline enables reliable automatic detection of human silhouettes and thwarts automatic detection of faces. We now investigate whether the pipeline sufficiently de-identifies humans to cause attempts at automated person identification to fail. We use the same 49 subject images and 1127 de-identified images described in Subsection 5.2.3. We assume that we have a perfect detector that correctly segments the

[3] Even though the CDNet 2014 dataset was selected because the image resolution is higher than in average surveillance datasets, in many frames the subject is simply too far from the camera for the face to be recognizable even to human observers.

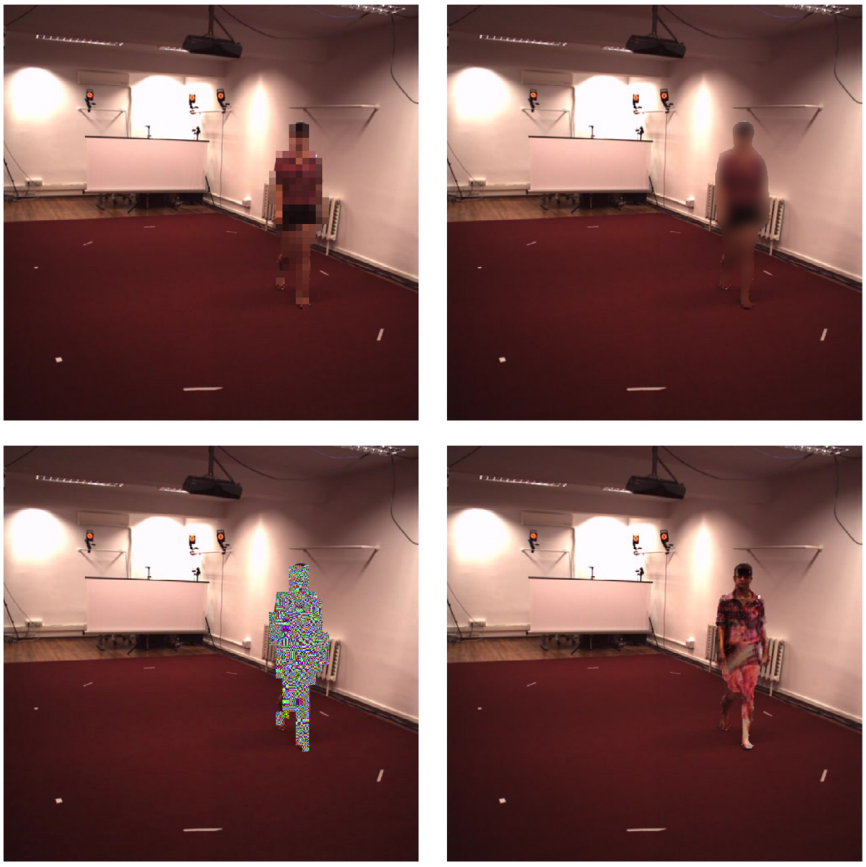

Fig. 12. Several examples of de-identification approaches: pixelization, blurring, DCT-block scrambling and the proposed method.

Table 2
Detector performance on original and de-identified images, in terms of detection accuracy and the average number of false positives per image.

Detector	Trained for	Accuracy [%]		FP rate (# per im.)	
		original	de-identified	original	de-identified
DPM	full body	88.0	83.4	0.22	0.25
ICF	full body	44.0	40.9	0.14	0.12
HOG	full body	81.6	75.9	0.52	0.45
Viola-Jones	upper body	79.6	70.6	0.44	0.51
Viola-Jones	frontal face	35.0	11.0	0.75	0.77

outline of the person in each image and build two databases of descriptors of segmented persons: (i) a database of persons as they appear in the original frames, and (ii) a database of de-identified persons. We then test whether the descriptors of de-identified persons can be automatically matched with the descriptors of the original persons, i.e. whether it is possible to determine the identity of the person using the de-identified image. We compare these findings with a baseline established on matching with person images that were not de-identified.

For person descriptors we use either 3D histograms of RGB, weighted histograms of per-pixel gradient orientations of the segmented persons, or a concatenation of both. The color histograms are built using 8 bins in three dimensions, resulting in a descriptor of size 256, while histograms of gradients are built using 16 bins. Matching the descriptors is performed using the k nearest neighbors algorithm (k-NN) using χ^2 as a distance measure (Altman, 1992). To find the identity of the person, the algorithm retrieves k database descriptors that are nearest to the query descriptor. The identity of the person is determined by finding the most common person in the retrieved k descriptors. If a tie happens (e.g. in 4-NN classification two retrieved descriptors belong to person A, and two to person B) the final identity is assigned by randomly selecting one of the candidate identities. In the case of the concatenated descriptor, the distance between two descriptors is computed as a sum of χ^2 distances between the two color histograms and χ^2 distances between the two gradient histograms.

Baseline classification accuracies obtained using leave-one-out cross-validation on the database of person descriptors from the original frames are shown in Fig. 13(a). Fig. 13(b) shows classification accuracies when matching de-identified person descriptors with the original person descriptors. In spirit similar to the baseline leave-one-out methodology, we remove the descriptor of the original person that generated the query de-identified person when searching for the nearest neighbors. Therefore, the classification accuracies are obtained under the assumption that the exact same image used as content for the de-identified image is not in the descriptor database.

We see that classification accuracies for the de-identified images are much lower than classification accuracies for the original images. While the original images can be classified with as high

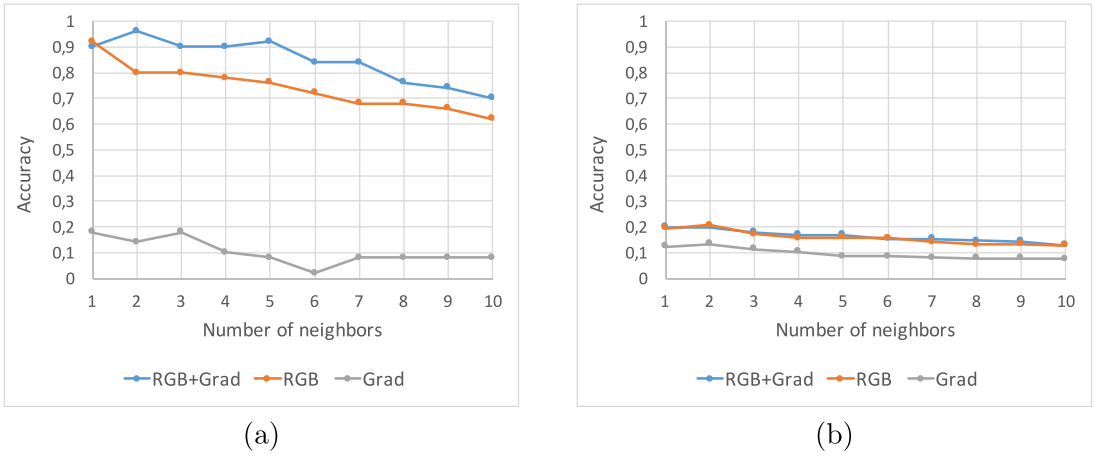

Fig. 13. The effects of de-identification on k-NN classification accuracy, depending on the number of neighbors k. (a) Classification accuracy of non-de-identified person descriptors, obtained using leave-one-out cross-validation on the descriptor database built from original images. (b) Classification accuracy when matching de-identified person descriptors with the descriptor database built from original images. Three descriptor types are shown: 3D RGB color histograms (RGB), histograms of gradient (Grad) and concatenation of both (RGB+Grad).

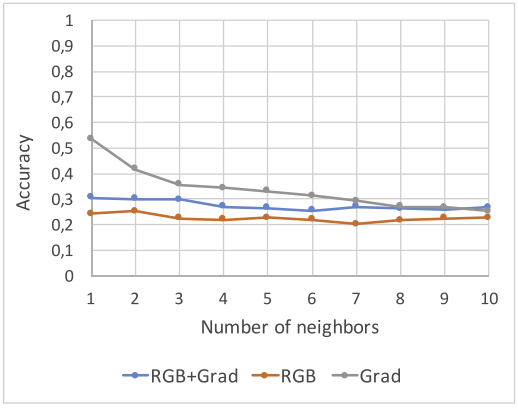

Fig. 14. The effects of de-identification on k-NN classification accuracy when matching de-identified person descriptors with the descriptor database built from original images, depending on the number of neighbors k. Three descriptor types are shown: 3D RGB color histograms (RGB), histograms of gradient (Grad) and concatenation of both (RGB+Grad). In contrast to Fig. 13 (b), here we allow matching the de-identified descriptor with the descriptor of the exact original image used to generate the de-identified image, resulting in a significant increase in performance for the gradient-based descriptor.

as 90% accuracy, the accuracy for de-identified images is around a constant 20% regardless of the value of k or the type of descriptor used. The descriptor based on histograms of gradients is performing particularly poorly. The reason for this is that the descriptor is very sensitive to the silhouette of the person (Dalal & Triggs, 2005), and in our database each subject is represented with an average of only five images filmed from different angles. As the original image is removed from the database in both experiments depicted in Fig. 13, there are almost no similar silhouettes in the database corresponding to the same person. In comparison, if the original image is retained, we see a large performance increase when using histograms of gradients, as shown in Fig. 14

The findings from this experiment indicate that our pipeline de-identifies the persons sufficiently to significantly decrease automated person identification.

5.2.5. A user study

In an approach complementary to automated evaluation of our de-identification pipeline, we perform a user study to evaluate how humans perceive and classify the de-identified images produced by the pipeline. The goals of the study are (i) to investigate whether humans are capable of correctly determining the identities of the de-identified persons, and (ii) to determine to what degree the soft and non-biometric identifiers in the de-identified images are obfuscated.

In the study, the user is first presented with a de-identified image and asked to find an image of the same person among five non de-identified choices. Next, the user is asked to classify individual soft biometric and non-biometric identifiers using the de-identified image. We repeat the process for six de-identified subjects and record the responses of 40 users. The results are summarized in Table 3, while the de-identified subjects are shown in Fig. 15.

It is interesting to note the high variance of user accuracy depending on the experiment. In some of the de-identified images it was very hard to guess the identity of the person, as well as qualify other features of the image (e.g. experiment 4), while for some images almost all users classified everything correctly (e.g. experiment 6). The degree of success of de-identification seems to be highly dependent on how the mixture of the original content image and the randomly selected style image turns out.

In summary, the user study has shown that humans are able to guess the identities of the persons de-identified by our pipeline rather well when presented with an original full-body image of the person. As the silhouette of the person remained preserved, the users often identified the person based on the body shape. When interacting with the users, we learned that sex was also inferred mainly based on the body shape, although hair length and clothing type were used as secondary clues. In a large scale application with a database of thousands of persons, we expect that identification by humans would be considerably harder than in this study where we had a database of only 10 persons with a lot of different body shapes.

Users often misclassified hair color and clothing color, indicating that these features are de-identified well. While the clothing type and hair length were classified with a relatively high accuracy, we believe that color is a much stronger cue than the type of clothing or hair length. Therefore, we believe that the performance of our pipeline is adequate for a reasonable level of de-identification.

Table 3
Recognition accuracy of various features in the user study for six de-identified images shown in Fig. 15, based on the responses of 40 users.

	Identification [%]	Sex [%]	Hair length [%]	Hair color [%]	Clothing type [%]	Clothing color [%]	Footwear type [%]
Experiment 1	92.5	90.0	45.0	47.5	50.0	5.0	40.0
Experiment 2	60.0	95.0	80.0	20.0	70.0	10.0	7.5
Experiment 3	92.5	89.7	87.2	45.0	77.5	27.5	72.5
Experiment 4	67.5	79.5	47.5	35.0	57.5	15.0	80.0
Experiment 5	67.5	95.0	90.0	32.5	95.0	35.0	82.5
Experiment 6	95.0	100.0	97.5	52.5	100.0	56.4	87.5
Total	79.2	91.5	74.5	38.8	75.0	24.8	61.7

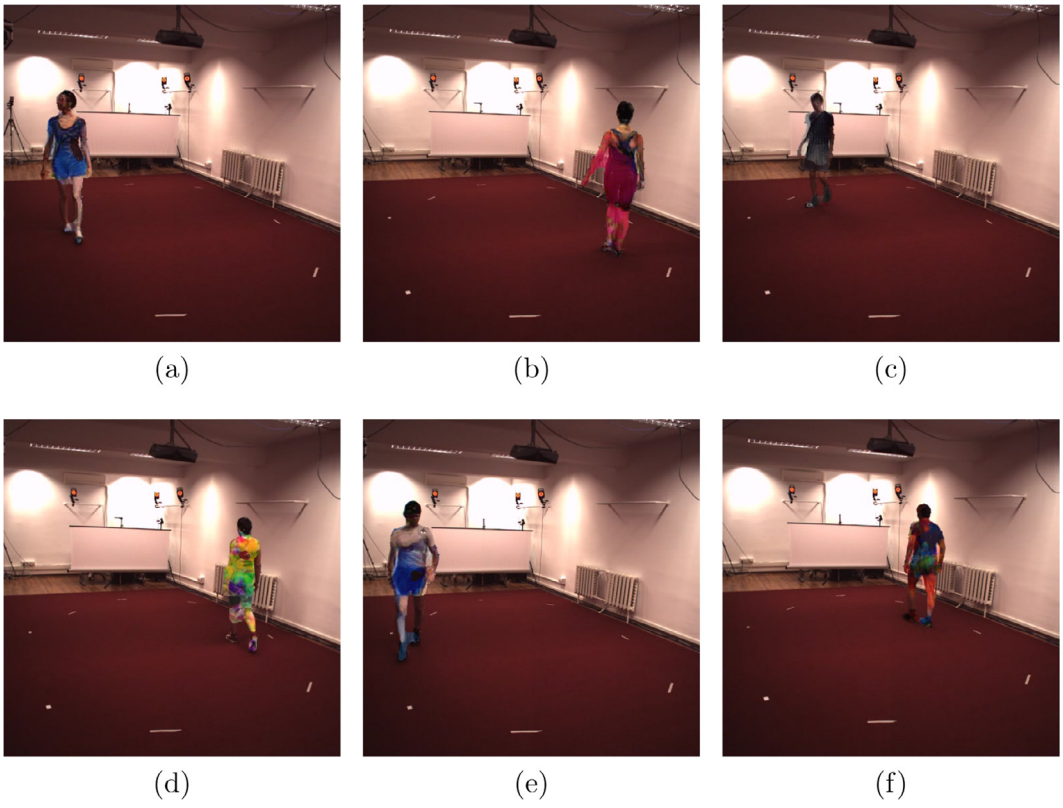

Fig. 15. The six de-identified images used in our user study. The letters a-f correspond to experiment numbers 1–6.

6. Conclusion

We have presented a computer vision-based pipeline for automated de-identification of humans in surveillance video sequences. The pipeline utilizes background subtraction based on Gaussian mixture models to obtain initial estimates of human locations, an improved GrabCut algorithm for precise human segmentation and the neural art algorithm to de-identify the segmented humans by altering their appearance. Experimental evidence suggests that the proposed pipeline successfully detects humans and produces highly accurate human segmentations. Through the application of the neural art algorithm, the appearance of the segmented humans is altered so that the detection of the faces of de-identified humans and the automated classification of their identity is unreliable. In a user study, we have shown that humans perceive a number of appearance features in the de-identified images as different than in the original images (e.g. hair and clothing color). Simultaneously, the silhouettes of the de-identified humans are preserved, helping maintain the naturalness of the scene.

Both our automatic evaluation and our user study have shown that body shape is a revealing feature that significantly helps to identify a person de-identified by our pipeline. Given a large enough database of persons, this problem should be somewhat mitigated, as there would be a number of persons with very similar body shapes. However, our future work will involve investigating whether body shape alterations can be incorporated into our pipeline while still maintaining the naturalness of the scene and the recognizability of what the person is doing.

Acknowledgements

This work was supported by the Croatian Science Foundation (DeMSI, UIP-11-2013-1544). This support is gratefully acknowledged.

References

Agrawal, P., & Narayanan, P. J. (2011). Person de-identification in videos. *IEEE Transactions on Circuits and Systems for Video Technology, 21*(3), 299–310. doi:10.1109/TCSVT.2011.2105551.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185. doi:10.2307/2685209.

Andriluka, M., Roth, S., & Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. *Ieee conference on computer vision and pattern recognition (CVPR)*.

Baltieri, D., Vezzani, R., & Cucchiara, R. (2014). Mapping appearance descriptors on 3d body models for people re-identification. *International Journal of Computer Vision, 111*(3), 345–364. doi:10.1007/s11263-014-0747-z.

Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2014). Ten years of pedestrian detection, what have we learned? *ECCV, cvrsuad workshop*.

Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., & Nayar, S. K. (2008). Face swapping: Automatically replacing faces in photographs. *ACM Trans. Graph., 27*(3), 39:1–39:8. doi:10.1145/1360612.1360638.

Blažević, M., Brkić, K., & Hrkać, T. (2015). Towards reversible de-identification in video sequences using 3d avatars and steganography. *CoRR, abs/1510.04861*.

Boykov, Y., & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer vision, 2001. ICCV 2001. proceedings. eighth IEEE international conference on: 1* (pp. 105–112vol.1). doi:10.1109/ICCV.2001.937505.

Bradski, G. (2000). The openCV library. *Dr. Dobb's Journal of Software Tools*.

Brkić, K., Hrkać, T., Sikirić, I., & Kalafatić, Z. (2016). Towards neural art-based face de-identification in video data. In *2016 first international workshop on sensing, processing and learning for intelligent machines (spline)* (pp. 1–5). doi:10.1109/SPLIM.2016.7528406.

Brutzer, S., Hoferlin, B., & Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the 2011 ieee conference on computer vision and pattern recognition*. In *CVPR '11* (pp. 1937–1944). Washington, DC, USA: IEEE Computer Society.

Cheung, S.-C. S., & Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. *Visual Communications and Image Processing 2004, 5308*(1), 881–892.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Mach Learn, 20*(3), 273–297. doi:10.1023/A:1022627411411.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. CVPR* (pp. 886–893).

Dollar, P., Tu, Z., Perona, P., & Belongie, S. (2009). Integral channel features. In *Proceedings of the british machine vision conference*. BMVA Press. doi:10.5244/C.23.91.

Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(4), 743–761. doi:10.1109/TPAMI.2011.155.

Dufaux, F., & Ebrahimi, T. (2008). Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology, 18*(8), 1168–1174. doi:10.1109/TCSVT.2008.928225.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1627–1645.

Garcia, J., Martinel, N., Gardel, A., Bravo, I., Foresti, G. L., & Micheloni, C. (2016). Modeling feature distances by orientation driven classifiers for person re-identification. *Journal of Visual Communication and Image Representation, 38*, 115–129. http://dx.doi.org/10.1016/j.jvcir.2016.02.009.

García-Martín, A., Cavallaro, A., Martínez, J., & Martínez, J. M. (2012). People-background segmentation with unequal error cost. In *2012 19th IEEE international conference on image processing* (pp. 157–160). doi:10.1109/ICIP.2012.6466819.

García-Martín, A., & Martínez, J. M. (2012). On collaborative people detection and tracking in complex scenarios. *Image Vision Comput., 30*(4–5), 345–354. doi:10.1016/j.imavis.2012.03.005.

García-Martín, A., & Martínez, J. M. (2015). People detection in surveillance: Classification and evaluation. *IET Computer Vision, 9*. 779–788(9)

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015a). A neural algorithm of artistic style. *CoRR, abs/1508.06576*.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015b). Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR, abs/1505.07376*.

Gross, R., Airoldi, E., Malin, B., & Sweeney, L. (2006a). In G. Danezis, & D. Martin (Eds.), *Integrating utility into face de-identification* (pp. 227–242)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gross, R., Sweeney, L., Cohn, J. F., De la Torre, F., & Baker, S. (2009). *Protecting privacy in video surveillance* (pp. 129–146)). Springer Publishing Company, Incorporated.

Gross, R., Sweeney, L., de la Torre, F., & Baker, S. (2006b). Model-based face de-identification. In *IEEE workshop on privacy research in vision, in conjunction with cvpr* (p. 161). doi:10.1109/CVPRW.2006.125.

Han, H., & Jain, A. K. (2013). Tattoo based identification: Sketch to image matching. In *Biometrics (icb), 2013 international conference on* (pp. 1–8). doi:10.1109/ICB.2013.6613003.

Heflin, B., Scheirer, W., & Boult, T. E. (2012). Detecting and classifying scars, marks, and tattoos found in the wild. In *Biometrics: Theory, applications and systems (btas), 2012 IEEE fifth international conference on* (pp. 31–38). doi:10.1109/BTAS.2012.6374555.

Hernandez-Vela, A., Reyes, M., Ponce, V., & Escalera, S. (2012). Grabcut-based human segmentation in video sequences. *Sensors, 12*, 15376–15393. doi:10.3390/s121115376.

Herrero, S., & Bescós, J. (2009). Background subtraction techniques: Systematic evaluation and comparative analysis. In *Proceedings of the 11th international conference on advanced concepts for intelligent vision systems*. In *ACIVS* (pp. 33–42). Berlin, Heidelberg: Springer-Verlag.

Hrkać, T., & Brkić, K. (2015). Iterative automated foreground segmentation in video sequences using graph cuts. In *Pattern recognition: 37th german conference, gcpr 2015, aachen, germany, october 7–10, 2015, proceedings* (pp. 308–319). Cham: Springer International Publishing. doi:10.1007/978-3-319-24947-6_25.

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(7), 1325–1339.

Kim, J., Parra, A., Yue, J., Li, H., & Delp, E. J. (2015). Robust local and global shape context for tattoo image matching. In *Image processing (icip), 2015 IEEE international conference on* (pp. 2194–2198). doi:10.1109/ICIP.2015.7351190.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (cvpr'05) - volume 1 - volume 01*. In *CVPR '05* (pp. 878–885). Washington, DC, USA: IEEE Computer Society. doi:10.1109/CVPR.2005.272.

Lin, Y., Wang, S., Lin, Q., & Tang, F. (2012). Face swapping under large pose variations: A 3d model based approach. In *2012 IEEE international conference on multimedia and expo* (pp. 333–338). doi:10.1109/ICME.2012.26.

Newton, E. M., Sweeney, L., & Malin, B. (2005). Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering, 17*(2), 232–243. doi:10.1109/TKDE.2005.32.

Ouyang, W., & Wang, X. (2012). A discriminative deep model for pedestrian detection with occlusion handling. In *Computer vision and pattern recognition (cvpr), 2012 IEEE conference on* (pp. 3258–3265). doi:10.1109/CVPR.2012.6248062.

Ouyang, W., Zeng, X., & Wang, X. (2013). Modeling mutual visibility relationship in pedestrian detection. In *Computer vision and pattern recognition (cvpr), 2013 IEEE conference on* (pp. 3222–3229). doi:10.1109/CVPR.2013.414.

Padilla-López, J. R., Chaaraoui, A. A., & Florez-Revuelta, F. (2015). Visual privacy protection methods: A survey. *Expert Systems with Applications, 42*(9), 4177–4195. http://dx.doi.org/10.1016/j.eswa.2015.01.041.

Park, S., & Trivedi, M. M. (2005). A track-based human movement analysis and privacy protection system adaptive to environmental contexts. In *Advanced video and signal based surveillance, 2005. AVSS 2005. ieee conference on* (pp. 171–176). doi:10.1109/AVSS.2005.1577262.

Poullot, S., & Satoh, S. (2014). Vabcut: A video extension of grabcut for unsupervised video foreground object segmentation. In *Proc. VISAPP*.

Reid, D., Samangooei, S., Chen, C., Nixon, M., & Ross, A. (2013). Soft biometrics for surveillance: An overview. In *Machine learning: Theory and applications*. In *31* (pp. 327–352). Elsevier.

Ribarić, S., Ariyaeeinia, A., & Pavešić, N. (2016). De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication, 47*, 131–151. http://dx.doi.org/10.1016/j.image.2016.05.020.

Rother, C., Vladimir, K., & Blake, A. (2004). "grabcut" - interactive foreground extraction using iterated graph cuts. In *Proc. SIGGRAPH*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *IInt. J. Comput. Vision, 115*(3), 211–252. doi:10.1007/s11263-015-0816-y.

Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In *Proc. CVPV* (pp. 3626–3633).

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Cvpr* (pp. 2246–2252).

Sun, J., Zang, W., Tang, X., & Shum, H.-Y. (2006). Background cut. In *Proc. CCCV* (pp. 628–641).

Viola, P., & Jones, M. (2001). Robust real-time object detection. *International journal of computer vision*.

Viola, P., Jones, M. J., & Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *Int. J. Comput. Vision, 63*(2), 153–161. doi:10.1007/s11263-005-6644-8.

Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014). CDnet 2014: An expanded change detection benchmark dataset. In *in proc. ieee workshop on change detection (cdw-2014) at cvpr-2014* (pp. 387–394).

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real–time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell., 19*(7), 780–785.

Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2016). How far are we from solving pedestrian detection? *Cvpr*.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Icpr (2)* (pp. 28–31).