Human segmentation in surveillance video with deep learning

Monica Gruosso[1] · Nicola Capece[2] · Ugo Erra[1] (ID)

© The Author(s) 2020

Abstract

Advanced intelligent surveillance systems are able to automatically analyze video of surveillance data without human intervention. These systems allow high accuracy of human activity recognition and then a high-level activity evaluation. To provide such features, an intelligent surveillance system requires a background subtraction scheme for human segmentation that captures a sequence of images containing moving humans from the reference background image. This paper proposes an alternative approach for human segmentation in videos through the use of a deep convolutional neural network. Two specific datasets were created to train our network, using the shapes of 35 different moving actors arranged on background images related to the area where the camera is located, allowing the network to take advantage of the entire site chosen for video surveillance. To assess the proposed approach, we compare our results with an Adobe Photoshop tool called Select Subject, the conditional generative adversarial network Pix2Pix, and the fully-convolutional model for real-time instance segmentation Yolact. The results show that the main benefit of our method is the possibility to automatically recognize and segment people in videos without constraints on camera and people movements in the scene (Video, code and datasets are available at http://graphics.unibas.it/www/HumanSegmentation/index.md.html).

✉ Ugo Erra
  ugo.erra@unibas.it

  Monica Gruosso
  monica.gruosso@unibas.it

  Nicola Capece
  nicola.capece@unibas.it

[1] Department of Mathematics, Computer Science, and Economics, University of Basilicata, Potenza, Italy

[2] School of Engineering, University of Basilicata, Potenza, Italy

⧸ Springer

1 Introduction

Fully-autonomous video analysis systems have become increasingly important in recent years [1, 21, 52]. The British Security Industry Association estimated that between 4.1 and 5.9 million closed-circuit televisions were installed in the UK in 2013. Both public and private surveillance systems around the world produce an enormous amount of data, which creates a challenge for big data and artificial intelligence. There are various applications related to intelligent video surveillance, such as human search, facial recognition, people counting, and vehicles detection. Today, traditional surveillance systems are being complemented and even replaced by advanced intelligent surveillance systems. These enable high accuracy monitoring, such as human activity recognition [40]. Human activity recognition system enables continuous monitoring of human behaviors in the area of surveillance, which allow tracking of human body parts such as head, torso, arms, and legs to perform activity recognition tasks.

In the surveillance systems, the background subtraction scheme for object classification is one of the first steps for human detection. Usually, a background subtraction algorithm captures a sequence of images containing moving humans from a static single-camera and detects them from the reference background image. Other methods use image segmentation to achieve the same purpose. There are two main classes of segmentation algorithms: instance and semantic segmentation. Approaches based on the first class of algorithms can identify specific regions that share similar features but do not interpret the content. Vice versa semantic segmentation takes care of understanding the content of the image by classifying the pixels and relating them to certain classes and is responsible for giving such content a closed and well-defined edge. It can be useful, for example, in scene understanding [3, 12, 64] or for medical purposes [2, 42]. Human segmentation is a subset of semantic segmentation, where the purpose consists of classifying the pixels in background and foreground.

In the scientific literature, there are various approaches related to human segmentation. Some of them are based on images captured by a static camera [62] or an infrared camera [17]. In other cases, there are no camera-related constraints, but additional data must be provided to the system to perform a good segmentation, such as pose [6] or contour information [54]. However, it is still a challenging task, especially if the purpose is to perform segmentation of the foreground automatically and without further information. Segmentation of people in a huge amount of videos, such as in video surveillance context, can benefit from deep learning techniques. The goal of deep learning is to develop computational models that consist of multiple processing layers used to learn data representations at multiple levels of abstraction [34]. The deep learning revolution has arisen due to the ability of certain computers to process data at nonlinear levels in a similar way to humans. In this way, the computers learn and perfect increasingly complex functionality.

In this paper, we propose an alternative approach for human segmentation in videos through the use of a deep convolutional neural network (CNN). The CNN has an encoder–decoder structure composed of two subnetworks: an encoding and a decoding component, appropriately trained to perform automatic extraction of actors from images. The encoder is a typical convolutional network and is topologically identical to the well-known VGG-16 architecture [50], but without the fully connected layers. The decoder has as many convolutional layers as those of the encoder, and converts the low-resolution encoder feature maps to full input resolution feature maps by using upsampling operations. In particular, this type of network classifies the pixels of the input images and produces an output image

segmented appropriately. Although some datasets including a person class exist, they do not contain a large number of high quality human segmentation masks. In similar tasks, such as image matting, datasets have only a few hundred unique foreground objects [58] or focus on portrait images [48]. To train our network, two specific datasets were created and combined with the background captured from the monitored area of the surveillance camera. We collected the shapes of 35 different moving actors and arranged them on background frames, which were extracted from videos recorded in selected surveillance sites: the first dataset concerned an indoor area and the second was related to an outdoor area. Actors were captured in different postures and distances, obtaining 16,832 training images with unique foreground shapes. The main benefit of our method is the possibility to automatically recognize and segment people in videos without constraints on camera and people movements in the scene. It can considerably support video surveillance systems that still require human supervision. Furthermore, our network output can also be used for other applications, for example, to quickly generate an accurate trimap on real images with people in the foreground.

Our main contributions include

- The attempt to reduce some constraints and overcome the challenges in the field of automatic human segmentation. In particular, limits of a static background, uniform lights and colors, camera and people movements in a specific area of interest.
- Indoor and outdoor scenes datasets creation with high-quality human segmentation masks.
- The separation of high and low frequencies in the images of our datasets using a nonlinear filter for better results.
- A preliminary study on the automatic segmentation of people in videos, testing a simple and efficient architecture in this field, and using the inverse frequency for class weighting in the network setup.
- Extensive quantitative and qualitative experiments to verify the proposed approach and prove its effectiveness.

The reminder of this paper is structured as follows: Section 2 provides an overview of related works; Section 3 provides background to semantic segmentation and CNN structures by describing the encoder and decoder; Section 4 describes our approach and the datasets used in the CNN training; Section 5 describes the CNN training phase; Section 6 summarizes our results and evaluation methods; Section 7 discusses comparisons with other approaches and potential additional applications of our work; Section 8 presents final remarks and the future direction of our research.

2 Related work

Increasingly, image segmentation techniques are being used to divide an image into a set of non-overlapping regions [5, 31, 39, 43, 49]. Many methods have been developed to tackle this task by applying it to medical image analysis [2, 28, 42], autonomous driving [15, 56], remote sensing [30], and video surveillance. In this latter case, automatic human segmentation can be very useful, especially in challenging conditions, where camera or person can move. Zhao et al. [62, 63] proposed a Markov chain Monte Carlo

approach to segment individual humans in a crowed scene acquired from a static camera. Fernández-Caballero et al. [17] presented a real-time people segmentation approach based on infrared images. Their algorithm starts applying traditional thresholding techniques to obtain candidate blobs, which are refined in a second phase. Another approach working with infrared images is W^4 system [22], a real-time visual surveillance system that detects people in an outdoor environment. W^4 detects foreground pixels with a statistical-background model and groups them into blobs that are classified using shape analysis. Vineet et al. [57] proposed a method for human segmentation in images and videos using CFR framework and included shape priors and histogram potential to help in recovering background regions and humans' shapes. Bhole et al. [6] used multiple CRFs and a pose detector, which helps guide the segmentation of challenging frames and obtain location information to refine the results. Other human segmentation methods are graph cut-based. Hernández et al. [25] presented a GrabCut methodology, where a HOG-based subject detection, face detection and skin color model were used for the algorithm initialization. Migniot et al. [38] introduced a graph cut weighted by a non-binary template useful to evaluate the shape of the silhouette. Furthermore, the segmentation was refined by a part-based template considering different postures of people. Song et al. [51] provided one of the first responses to the problem of fast human segmentation in an image-by-image manner. They tested some convolutional network architecture achieving a high acceleration of computing time and obtaining a decent segmentation accuracy. Tesema et al. [54] proposed a deep contour-aware network, which requires the mask and the contour to improve segmentation performance. The contour is generated using an edge detector and refined with some tricks not specified by the authors.

The main challenge in this scientific field is related to the lack of high quality labeled data [33]. In some cases, it is also necessary to calculate additional annotations to achieve a good result [6, 54, 61]. In other cases, there are strong limits: the camera is fixed [62] or moves slightly, the subjects must wear clothes of a different color from the background, which usually has a constant and uniform color, and shadows must be avoided [47]. It is still a challenge to be able to obtain a very accurate segmentation mask of a full-length subject, in real-time and in real-life situations. We propose an alternative automatic human segmentation method for supporting video surveillance systems. People can move in a chosen area, there are no camera constraints and no additional data is required in addition to the segmentation mask. The use of a deep neural network trained with a specific dataset guarantees a good level of accuracy.

3 Background

The proposed approach aims to recognize and segment people in images and videos. It can be very beneficial, for example, for supporting video surveillance systems. We use the deep neural network (DNN) shown in Fig. 1.

Our neural network is based on the well-known SegNet architecture [3], which is a CNN structured as an encoder–decoder, performing pixel classification through its final layer. SegNet provided good segmentation performances and proved to be efficient both in terms of computational time and memory. The encoding component of the input consists of several convolutional layers [20], with batch normalization operations [26], rectified linear units (ReLUs) as activation functions [19], and max-pooling layers [66]. The component that performs the decoding of the encoder output is based on inverse operations such as deconvolution [60] and unpooling [59].
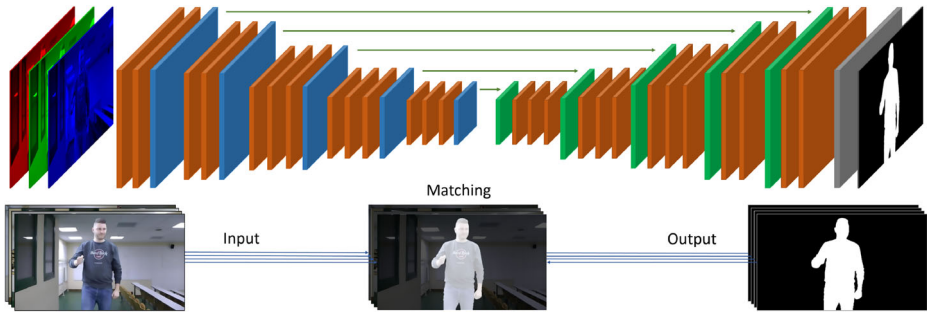
Fig. 1 The used encoder–decoder CNN, developed from the SegNet [3] architecture. The network structure is represented by colored blocks: orange represents the convolutions, batch normalization, and ReLU operations; blue represents the pooling operations; green represents the up-sampling operations; and gray represents the pixel classification layer based on Softmax operation. The set of images to the left are the input set, while the set of images to the right are the corresponding outputs

3.1 Encoder

The encoder is a subnetwork of our model, which consists of 13 convolutional layers based on VGG-16 CNN proposed by the Visual Geometry Group to solve classification problems [50]. This type of CNN was originally trained on 224×224 RGB images, which are pre-processed by subtracting the mean from the RGB computed on all training datasets. The input is passed through a stack of convolutional layers, which have a spatial resolution of 3×3, and stride and zero-padding set at 1. In this way, is possible to preserve the spatial resolution of the input, after the convolution operation. There are five max-pooling layers with a 2×2 kernel and stride of 2. Among the convolutional layers, the ReLU activation function is performed, which introduces nonlinearities over the hidden layers. The original VGG-16 had three fully connected layers [45] after the convolutional one and Softmax [7] as the activation function. Our encoder is similar to VGG-16, and the weights are initialized through the same pre-trained model obtained in the VGG-16 training step for the classification task [20]. Using pre-trained weights and/or layers, related to a network that solves a similar task, is a common practice in deep learning: This is useful for obtaining a starting point and making the training easier and faster. To train a network from scratch, for example using randomized initialized weights, takes much more time, and a large amount of data is needed. The fully connected layers of VGG-16 were removed, preserving only the convolutional layers in the encoder structure. This makes the encoder subnetwork smaller and easier to train than many other architectures [36, 41]. Each convolutional layer produces a set of features map called a tensor, which is normalized through a batch normalization and passed through the ReLU. Then, a max-pooling operation is performed to reduce the spatial factor of the activated tensor by 2. Before the max-pooling is performed, the indices of this operation are stored. Such indices represent the positions of the maximum value and will be used in the decoding phase to perform the up-sampling operations.

3.2 Decoder

The remaining part of the network performs the decoding of the encoder output tensor. This is another subnetwork, called a decoder, which has as many convolutional layers as those of

the encoder. Decoding occurs through a set of operations such as deconvolution and unpooling. The unpooling operation performs the spatial up-sampling by using the stored indices in max-pooling operations. In this way, the hardware memory is significantly reduced, avoiding the need to store all of the features map, which was obtained after the down-sampling operation in the encoding phase, and reducing the number of parameters without loss of accuracy. The features maps obtained in this phase are sparse, and they are then passed through convolutional layers to make them dense. After the convolution, a batch normalization is performed. The last layer of the decoder network classifies pixels by using Softmax as the activation function. The loss function uses cross-entropy [20], which is typically used in classification problems. Weights of the decoder network are initialized using the MSRA method, which was proposed by He et al. [24]. It is suitable for layers followed by a ReLU activation function on very deep neural networks.

4 DNN for human segmentation

To train our DNN, we provided a pair of images as input. Each pair consists of an RGB element, which contains an actor's shape on a background frame, and a binary image, which represents the label (the segmentation mask) as shown in Fig. 1. The background frames were extracted from a recorded video in defined areas. In this way, once the neural network has been trained, it will be possible to use as input the frames of a video shot in the chosen area in which both the actor and the camera can move.

The RGB element is represented through a tensor $w \times h \times 3$, with width w, height h, and depth 3. The label is represented through a binary matrix that uses only two values, 0 and 1, where 0 represents the background class and 1 represents the foreground class. More specifically, the label represents the ground truth to be compared with the DNN output in order to compute the loss value during the training and validation phases.

Loss function was defined using cross-entropy, as indicated above. In particular, the error $E(w_i)$, related to the ith weight, can be defined as follows:

$$E(w_i) = -t^{(i)} \log(y^{(i)}) \tag{1}$$

where y is the prediction and t is the target related to foreground and background classes.

Cross-entropy is useful for measuring the dissimilarity between the ground truth and the output predicted by our network. Equation (1) can be defined as

$$E(w_i) = -t_{fg}^{(i)} \log(y_{fg}^{(i)}) - t_{bg}^{(i)} \log(y_{bg}^{(i)}) \tag{2}$$

where

- y_{fg} is the probability of classifying the output as foreground and y_{bg} is the complementary probability of classifying the output as background;
- t_{fg} is the real probability related to foreground and t_{bg} is the complementary real probability related to background.

Using (1) and (2), the loss function $E(w)$ computed on N samples and expressed according to the foreground terms, can be described as follows:

$$E(w) = \frac{1}{N} \sum_{i=1}^{N} \left[-t_{fg}^{(i)} \log\left(y_{fg}^{(i)}\right) - \left(1 - t_{fg}^{(i)}\right) \log\left(1 - y_{fg}^{(i)}\right) \right] \tag{3}$$

If an image contains only one actor, in general, the background pixels will be more frequent with respect to the foreground ones, as shown in Fig. 2. Such difference can be dangerous for the training process due to partial learning that favors the background class. Ideally, we would like each class to have the same number of observations in the training dataset to prevent a category from being underrepresented. Since the goal of the network is to segment actors in the foreground, the inverse frequency is used to weigh the classes and give more importance to the foreground. Thus, during back-propagation, only the gradient from the maximally scoring instance is calculated and used for updating the weights. The inverse frequency can be expressed as follows:

$$F^{-1} = \frac{\sum_{i,j} I_{i,j}}{\sum_{i,j} I_{i,j}^{(k)}} \qquad k = 0, 1 \tag{4}$$

where the numerator represents the sum of the pixels for each I image in the training dataset, and the denominator represents the sum of the pixels belonging to the kth class for each I image in the training dataset. F^{-1} is the inverse frequency for each class (background when k is equal to 0, foreground when k is equal to 1).

4.1 Dataset

Although there are some dataset including a person class, such as Pascal VOC [16], MS COCO [35] and the new YouTube-8M Segments, large dataset with high quality human segmentation masks are scarce. Recently, Supervisely Person Dataset has been released. It consist of 5711 images with unique people in the foreground. There are some accurate and high resolution masks. Unfortunately, there are not so many. Most were obtained through polygons and some of them are very coarse. Furthermore, in some cases, people cover small parts of the image.

To train our network, we collected a large number of unique human images with corresponding segmentation masks. 35 people were captured in different postures, such as

Fig. 2 A label overlay of a training image. The background pixels (light blue), are more frequent with respect to the foreground pixels (red). The frequency of foreground pixels in the training dataset is 11.16%, and the frequency of background pixels is 88.84%

walking, turning on themselves, jumping, greeting, and different distances obtaining, for example, full-length or half-length images.

Our dataset was based on a number of requirements to make a good dataset for semantic segmentation: *(i)* There must be enough image pairs, composed of those to be segmented and the relative ground truth, and *(ii)* the labels must be as precise as possible, due to potentially critical issues such as the actor's hair is very jagged, clothes have folds, or there is little difference between background and actor.

The first step in creating the dataset was to recorder videos of moving actors using the green screen and a camera to obtain a large amount of data and very precise labeling. The green screen setup consisted of an opaque green drape and two lights, one of which illuminated the background, and the other illuminated the actor to remove the shadows as much as possible. The videos were recorded using a Panasonic HDC-SD800 camera with 14.2 megapixel and $1,920 \times 1,080$ spatial resolution. The video preprocessing phase consisted of removing the green background (left panel of Fig. 3) using Adobe After Effects and extracting the shapes of the actors. Using the same software, it was also possible to obtain a matte version of each shape (center panel of Fig. 3). This consists of a mask that defines the transparent or background areas as black and the matte or foreground areas, which contain the actor's shape, as white. The matte version contains three channels, between 0 and 255. To label (right panel of Fig. 3) each image, a binarization process was performed using a global threshold value, which creates a binary image in which all the values of the starting image over the threshold are set to 1 and all the values below it are set to 0. The threshold is a value that varies between 0 and 1, but we set this value to 0.3 because this represented a good compromise for maintaining, as much as possible, the quality of the matte version. Lower values of the threshold would have made the background areas white, and higher values would have cut out part of the shape, such as edges, hair, and clothes.

The second step was to extract background frames from a recorded video in an indoor area that we chose for the video surveillance system. Then, the actors' shapes were arranged on the background frames to create the RGB elements. In this way, the network takes advantage of the knowledge of the area that the camera is able to frame. The use of many unique shapes and the balance of weights explained in Section 4 allows the network to not over-adapt.

The obtained pairs of images and labels were divided into 16,832 for the training set, 1,403 for the test set, and 900 for the validation set. The validation set was introduced to check the training performance of the network by using different data from the training set.

Fig. 3 Example of pre-processing phase. Left: The input taken using a camera and green screen. Center: The edge highlighted on the matte image obtained using Adobe After Effects. Right: The corresponding label

The dataset images were reduced using 640×360 spatial resolution for reasons related to the resources and computational times that were available.

Subsequently, we realized some critical issues that arose in this dataset solution. In particular, shape overlapping on the background frame caused a clear distinction between the borders of the shape and the background compared with a real image. The network learned the shape from images created in this manner, and didn't dive such good results on real images where the variation between the background and the shape was more linear.

To solve this problem, we introduced a bilateral filter [55] and applied it to the images obtained in the previous solution. This is a nonlinear filter and is often used to reduce the noise of images while preserving the edges. In practical terms, a weight is assigned to each pixel of the image to be filtered, depending both on spatial proximity (spatial domain) and on photometric similarity (range domain or intensity domain). The idea behind many spatial filters is based on the requirement that neighboring pixels tend to have similar values. However, this idea turns out to be incorrect on the borders of objects in images because, in these points, the signal changes quickly. The bilateral filter considers this feature and replaces the intensity of each pixel with a weighted average of the intensity value of the neighboring pixels. In this way, some of the high frequencies were removed, which are present in particular on the edges of objects, retaining the low frequencies. In realistic photos, normally, there is no clear separation between the actor and the background. Since our photos resulted from the overlapping of background frames and actors' shapes clipped from the green screen, this separation was more evident in the image in the left panel of Fig. 4. The bilateral filter was applied in the image in the right panel of Fig. 4, with the spatial parameter σ_d set to 16, the range parameter σ_r set to 0.1, and the dimension of a half-window of the Gaussian kernel w set to 5.

These parameters are inspired by those proposed by Durand and Dorsey [14] to separate low and high frequencies through the bilateral filter. Although they propose the σ_d value has to be equal to the 2% of the maximum dimension of the image, we increase this value

Fig. 4 Comparison between an unfiltered image and the corresponding image filtered with the bilateral filter. The filter parameters are $w = 5$, $\sigma_d = 16$, and $\sigma_r = 0.1$. In the filtered image, an attenuation of the sharp edges can be seen, particularly at the neck and the hands

to a 2.5% because in this way the quality of the approximation is experimentally improved for the used context [4]. Moreover the value of σ_r was selected as 0.1 based on visual effect of the the filter on the image and on the behaviour of the deep neural network. From our experiments, we established that these values are a good choice for our aim [9].

Based on the knowledge acquired, we also used the filtered dataset solution to build an outdoor scenes dataset. In this case, the second step was performed by extracting the background frames from a recorded video in an outdoor area.

Data augmentation was applied to the both indoor and outdoor datasets [13]. This operation generated perturbed images of the training dataset for each epoch, to avoid the problem of overfitting. The data augmentation was performed by applying several transformations on the fly in the training phase; therefore, the perturbed images were not stored, and the dataset dimension remained unchanged. We made spatial transformations mirroring right/left and rotation with a random angle between −30 and 30 degrees.

5 Training phase

We developed, trained, and tested our neural network using MATLAB® and its toolboxes, in particular, the Neural Network Toolbox™. We used Nvidia GPU GTX 1080 Ti, which has a Pascal architecture, 3584 CUDA core, 11 GB GDDR5X as a frame buffer and 11 Gbps speed memory. The training was performed by selecting ADAM [32] as the optimization algorithm. It was demonstrated empirically that this algorithm achieves good results, in a short time, applied to large models and datasets. The initial learning rate was set to 10^{-5}, and the validation set was introduced to check the level of generalization of the neural network for each epoch. The number of epochs was set to 40, but the training was stopped after 19 epochs since no further improvements were noted: neither an increase in accuracy nor a reduction in error. At the end of training, the accuracy was fixed at around 99.8%, and loss in training was around 0.01. Accuracy in the validation stage did not decrease with respect to that of training, reaching approximately 99.79%, and loss during validation fell throughout the training, reaching a final value of 0.008.

6 Results

Our model inference was performed by using one Titan Xp GPU card. Images with size 640×360 were processed in about 0.06 s. A video demo[1] of our work shows the results obtained interactively on videos recorded with different frame rates and in different locations. It can be seen that there are no constraints on the movement of the camera or people.

The remainder of the section shows an overview of the evaluation methods of the results obtained and analyzes the output related to both the test dataset and the photos taken in the areas chosen during the dataset creation phase.

6.1 Evaluation methods

Evaluation of the results after several training steps was performed through two types of test: A first test was performed by using the test set, which contained the images to be segmented

---

[1]URL http://graphics.unibas.it/www/HumanSegmentation/index.md.html

Table 1 The metric values of the whole test dataset for the first two trained model: Global Accuracy, Mean Accuracy, Mean IoU, and Mean BF score

	Global Acc.	Mean Acc.	Mean IoU	Mean BF Score
1st Network (Unfiltered Case)	0.99737	0.99828	0.99132	0.9942
2nd Network (Filtered Case)	0.99778	0.99829	0.99267	0.9955

The first row is related to the network trained with the unfiltered indoor scenes dataset, whereas the second row is related to the filtered indoor images

and the labels to be used as a comparison with the output obtained from the network; the other test was performed on real and unlabeled photos. Semantic segmentation quality was evaluated through three metrics: Accuracy, Intersection over Union (IoU) [18], and Mean Boundary F1 (BF) Score [46].

The Accuracy metric measures the amount of correctly classified pixels with respect to the total amount of pixels. It can represent the ratio of correctly classified pixels to total pixels, regardless of class (Global Accuracy), the ratio of correctly classified pixels in each class to total pixels, averaged over all classes (Mean Accuracy), or the ratio of correctly classified pixels in each class to the total number of pixels belonging to that class according to the ground truth. This last definition can be expressed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP indicates the true positive, TN indicates the true negative, FP indicates the false positive, and FN represents the false negative.

IoU is a statistical measure of accuracy that penalizes false positives. This parameter shows the quality of the pixels correctly classified with respect to the total amount of pixels assigned to a certain class by ground truth and by the network output. IoU can be expressed by the following formula and can also be computed as an average value (Mean IoU).:

$$IoU = \frac{TP}{TP + FP + FN} \tag{6}$$

BF Score is a measure of the accuracy used in the statistical analysis and is calculated for each class. The measure takes into account the precision and recovery of the test, where the precision is the number of true positives divided by the number of all positive results, and the recovery is the number of true positives divided by the number of all the tests that should have been positive (i.e., the sum of true positives and false negatives). This parameter is defined as the harmonic mean of precision p and recovery r.

Table 2 The metrics obtained by considering each class with respect to the unfiltered and filtered cases: Accuracy, Intersection over Union, and Mean BF score

		Acc.	IoU	Mean BF Score
1st Network	Foreground	0.99974	0.98587	0.99168
(Unfiltered Case)	Background	0.99683	0.99677	0.99672
2nd Network	Foreground	0.99909	0.98805	0.99351
(Filtered Case)	Background	0.99749	0.99728	0.9975

Table 3 The confusion matrix related to the test carried out on the networks after training with indoor scenes dataset

	1st Network (Unfiltered Case)		2nd Network (Filtered Case)	
	Predicted Foreground Class	Predicted Background Class	Predicted Foreground Class	Predicted Background Class
True Foreground Class	99.97	0.02647	99.91	0.09141
True Background Class	0.3166	99.68	0.2514	99.75

$$BF\ score = 2 \cdot \frac{p \cdot r}{p + r} \tag{7}$$

In addition to these metrics, a further method of viewing the performance data from the tested network is the normalized confusion matrix [53]. It returns a representation of the accuracy of the classification. Each column represents the predicted values, and each row represents the real values. Each element (i, j) is given from the amount of pixels that belong to the true class i, but associated with the predicted class j. A normalization is performed by dividing by the total number of predicted pixels in j.

6.2 Results after training with the indoor scenes dataset

The first two network training were performed respectively with the unfiltered and filtered dataset, both relating to an indoor area. The neural network performance was evaluated against the test set and the real photos. The test set was created as the training set, with actors' shapes arranged on background frames. Table 1 shows the metrics aggregated over the test dataset, and Table 2 shows the metrics for each class. Similar values were obtained for the tests carried out on the first model, 1st Network (Unfiltered Case), and on the second model, 2nd Network (Filtered Case). By observing the normalized confusion matrix (Table 3), it is possible to obtain rapid feedback on the performed test.

Figures 5 and 6 show some results concerning the first test phase performed respectively with the unfiltered and the filtered dataset. They highlight the best and the worst

Fig. 5 Results related to the test dataset after training using the unfiltered dataset. The test set was created as the training set and contains unfiltered images. The top left panel shows the best result for Mean Accuracy (99.9%), and the top right panel shows the best result for Mean IoU (99.6%). A classification error related to foreground can be seen in the bottom left panel, which shows the worst result for Mean Accuracy (96%), and a classification error related to background can be seen in the bottom right panel, which shows the worst result for Mean IoU (96.6%)
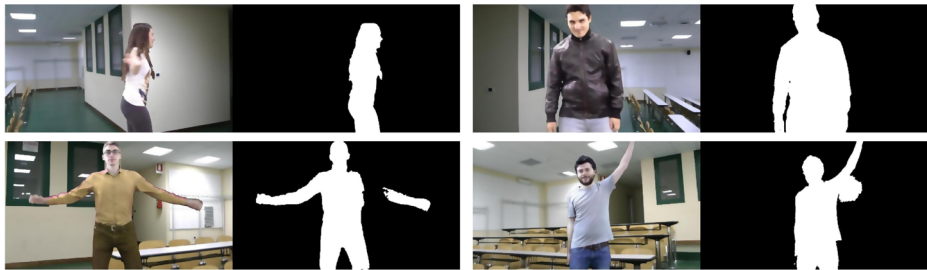
Fig. 6 Results of the filtered test dataset for the chosen indoor area. The top left panel shows the best results for Mean Accuracy (99.9%), and the top right panel shows the best results for IoU (99.6%). A classification error related to foreground pixels can be seen in the bottom left panel, which shows the worst result for Mean Accuracy (96.4%), and a classification error related to background pixels can be seen in the bottom right panel, which shows the worst result for Mean IoU (94%)

case for Mean Accuracy (left panels) and Mean IoU (right panels). As noted previously by analyzing the metrics, there is not a big difference between the two networks in the first test phase.

The second test phase concerned the evaluation of real photos, which were taken in the same indoor area where the background videos used to create the dataset were shot. These photos were resized appropriately to fit the aspect ratio to the size of the neural network input. Tables 4 and 5 show the quantitative results obtained by querying the network trained with the unfiltered dataset (1st Network) and the network trained with the filtered dataset (2nd Network). The metric values demonstrate a significant improvement in the case of the 2nd Network. In particular, the values for the foreground class increase by about 10% for Accuracy and IoU, and by 7% for Mean BF Score (Table 5). Figure 7 shows the qualitative results for both the networks. The 1st Network partially identified the subjects in these photos. Since the network was trained on a dataset in which the actors presented a well-defined edge with respect to the background, there is an uncertainty in real cases on the boundary zones. This involves an error in the classification of foreground pixel. As can be seen in the third panel of Fig. 7, we found a marked improvement in the classification of foreground pixels with the second network. The subjects were clearly identified in the first two cases, although the left hand was not correctly segmented in the third case. This latter is a challenging picture because the hand covers a small part of the image and has a color similar to the background.

Table 4 The metric values of the second test phase concerning the evaluation of real photos taken with a camera in the chosen indoor area

	Global Acc.	Mean Acc.	Mean IoU	Mean BF Score
1st Network (Unfiltered Case)	0.97937	0.93016	0.91083	0.92538
2nd Network (Filtered Case)	0.99289	0.98649	0.96947	0.97509

The first row is related to the network trained with the unfiltered indoor scenes dataset, whereas the second row is related to the network trained with the filtered images

Table 5 The metrics obtained by considering each class and related to the second test phase for the indoor area

		Acc.	IoU	Mean BF Score
1st Network	Foreground	0.86364	0.84491	0.89074
(Unfiltered Case)	Background	0.99668	0.9767	0.96001
2nd Network	Foreground	0.97784	0.94708	0.96487
(Filtered Case)	Background	0.99514	0.99185	0.98531

6.3 Results after training with the outdoor scenes dataset

The final training step was performed on the filtered dataset with background frames of an outdoor area, and the neural network performances were evaluated as in the previous cases. Table 6 shows the metrics aggregated over the test dataset, Table 7 shows the metrics for each class, and Table 8 shows the normalized confusion matrix.

Figure 8 shows some results concerning the first test phase performed with the network trained with the outdoor scenes dataset: the best and the worst case for Mean Accuracy (left panels) and Mean IoU (right panels). Figure 9 shows some of the second test phase results. They were obtained from the real photos taken in the same outdoor area selected for the network training.
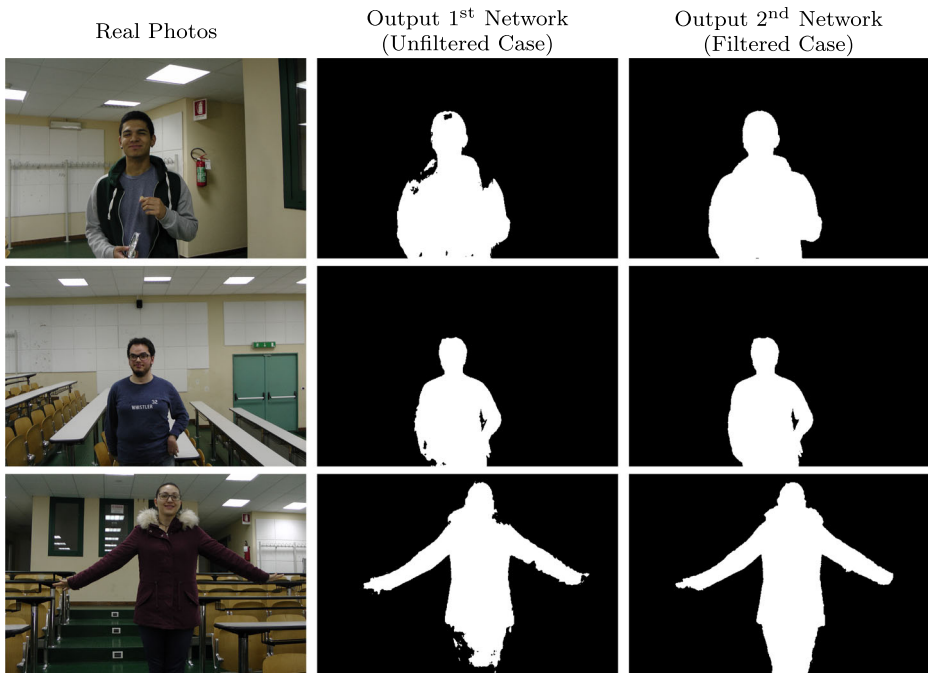
7 Comparisons and applications

Starting with the results obtained through our approach, we made some quantitative and qualitative comparisons with other approaches.

The first comparison was made using an Adobe Photoshop tool called Select Subject. It was developed through Adobe Sensei, a framework that uses artificial intelligence to support image-processing tasks to enhance and simplify complex user image-editing operations. According to Adobe, this tool is useful for quickly selecting prominent subjects in pictures. Select Subject provides a basic method to select the subject and allows the selection to be refined through other tools or user actions. Adobe Sensei is an advanced machine learning technology trained to identify a wide variety of objects in an image, such as people, animals, vehicles, and toys.

The second comparison was made against Pix2Pix, a type of Conditional Generative Adversarial Network developed by Isola et al. [27]. The authors tested their approach on several tasks, such as photo generation, image colorization, and segmentation. We trained Pix2Pix[2] using our dataset and the training information provided for the *Cityscapes labels* → *photos* task. In particular, the model was trained from scratch initializing the weight with a Gaussian distribution with zero mean and standard deviation 0.02. We set the number of epochs to 200 and the batch size to 10.

Finally, we focused on Yolact, a fully-convolutional model for real-time instance segmentation developed by Bolya et al. [8]. The authors achieved good performance training

---

[2]Implementation of Pix2Pix: https://github.com/affinelayer/pix2pix-tensorflow

| Real Photos | Output 1st Network (Unfiltered Case) | Output 2nd Network (Filtered Case) |

Fig. 7 Results of real photos, taken directly with a camera, querying the network after training with the unfiltered and filtered dataset. A classification error related to foreground pixels can be seen in the second column. In particular, parts of the actor in the top and bottom panels have a color similar to the background. The third column shows the results in the filtered case. We can see an improvement in the classification of foreground pixels, especially in the top panel

and testing their network on MS COCO dataset [35]. We conducted two assessments:[3] *(i)* we trained Yolact on our dataset as proposed by the authors, using batch size 8 and pre-trained weights provided for this model; *(ii)* we tested their base model, Yolact-550 with ResNet-101 backbone.

7.1 Quantitative result on test dataset

Firstly, we made a quantitative comparison calculating metrics value on the test dataset for Pix2Pix and Yolact, since we trained those networks with our indoor scenes dataset. Tables 9 and 10 illustrate the results on the whole test set and for each class, respectively. We found a decrease from 1% to 3% of the values achieved by the compared approaches, as demonstrated by all the metrics in Table 9 and foreground class metrics in Table 10.

In addition, we made a further comparison considering inference time per image, as illustrated in Table 11. The second column indicates the type of architecture of each neural network and the third one contains the inference time in seconds. For each approach, it was calculated on the same computer using the Titan Xp GPU card and after loading the neural

---

[3]For the first comparison, we downloaded the ImageNet pretrained model `resnet101_reducedfc.pth` and trained it using the information provided on the github page https://github.com/dbolya/yolact and our indoor scenes dataset. For the second evaluation, we used the Resnet101-FPN model, with input image size 550.

Table 6 The metric values of the whole test dataset for the network trained with our filtered dataset for the chosen outdoor area

Global Acc.	Mean Acc.	Mean IoU	Mean BF Score
0.99771	0.99843	0.99245	0.99602

network into memory. Therefore, the inference time includes the average time of reading and loading an image and the forward step to obtain the prediction. Although the inference time for the other approaches is less, we still got an interactive response with our approach via MATLAB. Note that computational times can be further reduced by generating C and CUDA code or MEX functions from MATLAB code.

7.2 Quantitative results on real photos

We compared all approaches using real photos. This test provided us with initial feedback on data that differs from training and test dataset. We considered a representative set of images, which consists of photos taken with a camera in the indoor area chosen for the training of our neural network. We needed ground truth images for each photo to calculate metrics. We got them through careful manual segmentation. Since it is time-consuming and, in some cases, it is not easy to obtain, we decided to consider 30 photos containing people at different distances from the camera and in several positions. Table 12 shows the metrics aggregated over the whole set of images and the best values for each metric are in bold. Although Photoshop Select Subject achieved better values for Global and Mean Accuracy and Mean IoU, our approach remained competitive: a very small gap for these metrics and the highest value for the Mean BF Score can be seen. Table 13 indicates the metrics for each class. The best values for the foreground class are highlighted in bold. As can be seen, our network achieved the highest value for Mean BF Score and differed slightly from the best Accuracy and IoU. Interestingly, Yolact-500 achieved excellent results in the case of Accuracy and the IoU for background class predictions, but not for the foreground class, with a difference greater than 20%. A high value for the background class may not always indicate a better overall result. Background pixels are more frequent than foreground pixels if the image contains a person distant from the camera, as indicated in Section 4.

7.3 Qualitative results on real photos

A qualitative and visual comparison of the output masks allowed us to have comprehensive and clear feedback on the performance of all methods. The third row of Fig. 10 shows some of the results obtained from real photos using Select Subject. The foreground shapes of its output were highlighted in white. In comparison with our output (second

Table 7 The metrics obtained by considering each class with respect to the filtered test dataset for the chosen outdoor area

	Acc.	IoU	Mean BF Score
Foreground	0.99957	0.9877	0.99432
Background	0.99729	0.9972	0.99771

Table 8 The confusion matrix related to the test carried out on the network after training with the filtered dataset for the chosen outdoor area

	Predicted Foreground Class	Predicted Background Class
True Foreground Class	99.96	0.04345
True Background Class	0.2705	99.73

row), less accuracy was obtained in the case of Photoshop, particularly in the first panel and the subject's contours in the third panel. The fourth row shows the output obtained in the case of Pix2Pix. A deterioration when using this model compared with the results of our network can be observed, mostly in the first two Pix2Pix output masks, in which the foreground is not completely identified or errors are made in the background. The bottom panels of Fig. 10 provide the output masks for both Yolact evaluations. The Yolact network trained with our dataset provided coarse segmentation masks, while Yolact-500 got better masks from the second and third photos, although no people were detected in the first photo.

All these tests show the effectiveness of the proposed work in terms of foreground and background accuracy and quality of the segmentation mask.

7.4 Other applications

The output of our neural network can be used for other applications, such as image-matting. In particular, a good trimap can be obtained from our segmentation mask. Most state-of-the-art matting algorithms [10, 11, 23] require human intervention to generate the alpha matte from the input image. The most common form of user interaction is the trimap interface, where the user manually partitions the image into foreground, background, and unknown regions [44]. Natural-image matting is usually problematic. User-specified strokes or trimaps are used to sample foreground and background colors to make it tractable. Our approach uses semantic segmentation that takes only an RGB image as input and generates a binary output quickly, often with a very accurate boundary. Therefore, our estimated segmentation result could be used as a good initial trimap. The unknown regions could be created with morphological operations such as erosion and dilation of foreground regions, as shown in Fig. 11. The trimap (top right panel) was generated using morphological
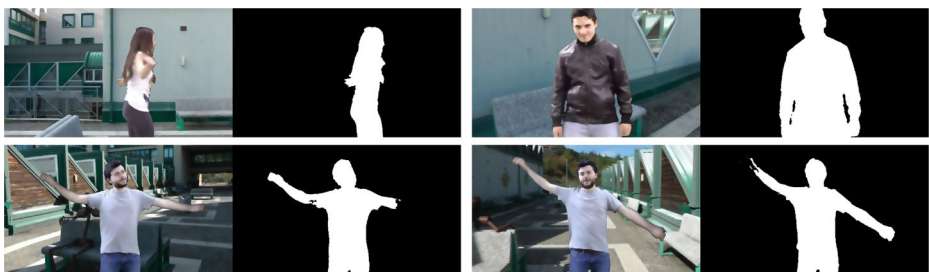
Fig. 8 Results of the filtered test dataset for the chosen outdoor area. The top left panel shows the best results for Mean Accuracy (99.9%), and the top right panel shows the best results for IoU (99.66%). The bottom left panel shows the worst result for Mean Accuracy (96.6%), and the bottom right panel shows the worst result for Mean IoU (97.8%). In these last two panels, a classification error related to foreground pixels can be seen
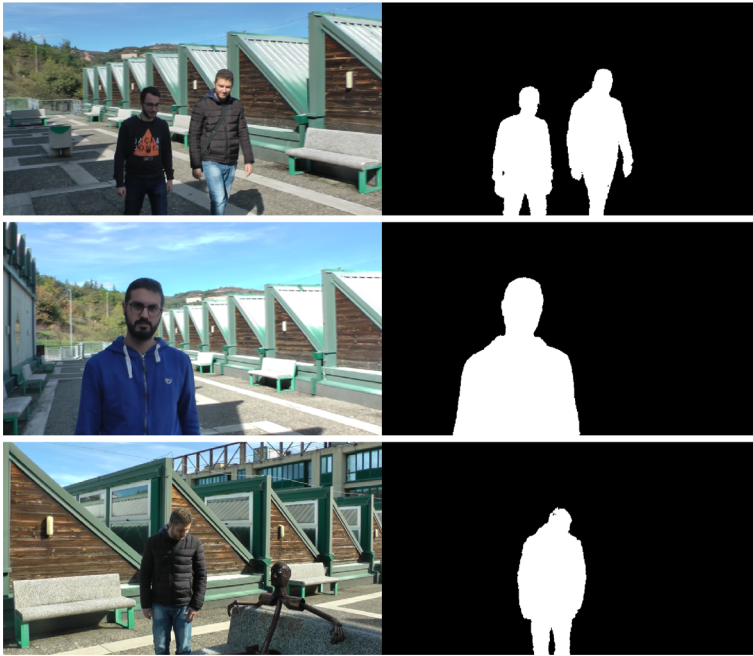
Fig. 9 Results of real images, taken directly with a camera, querying the network after training with the filtered dataset for the chosen outdoor area. There are no notable classification errors. The system is efficient even when there are more people in the image, as shown in the top panel

operations: the unknown region, in gray, was obtained with erosion and dilation using a disk-shaped structuring element with a radius of 20 pixels. The bottom panels show the KNN (K-Nearest Neighbors) [10] and the KL-divergence (Kullback-Leibler) [29] output using our trimap.

8 Conclusions

We aim to automatically segment people in videos using a deep convolutional neural network. People and camera can move in a specif area, and no additional data is required. The developed approach can support intelligent surveillance systems during the background subtraction step for human segmentation.

Two case studies were defined: human segmentation in indoor and outdoor environments. The semantic segmentation problem was tackled by classifying pixels and assigning

Table 9 The metrics values calculated on the test dataset. We tested our approach, Pix2Pix, and Yolact trained with our indoor scenes dataset

	Global Acc.	Mean Acc.	Mean IoU	Mean BF Score
Our Network	**0.99778**	**0.99829**	**0.99267**	**0.9955**
Pix2Pix	0.98579	0.9853	0.96827	0.96619
Yolact	0.99272	0.98597	0.97604	0.97534

The highest values are highlighted in bold

Table 10 The metrics for each class considering the test dataset

		Acc.	IoU	Mean BF Score
Our Network	Foreground	**0.99909**	**0.98805**	**0.99351**
	Background	0.99749	0.99728	0.9975
Pix2Pix	Foreground	0.98389	0.95741	0.95616
	Background	0.9867	0.97912	0.97623
Yolact	Foreground	0.9753	0.96095	0.96511
	Background	0.99664	0.99113	0.98558

Both Pix2Pix and Yolact were trained with our indoor scenes dataset

The highest values for the foreground class are highlighted in bold

them a specific meaning using the encoder-decoderr neural network. Our system identifies pixels belonging to the background class and those belonging to the foreground class, where the foreground is represented by the shape of a person in the image. The system also proved to be efficient when a photo contains more than one person, as shown in the top panel of Fig. 9. A critical issue that was tackled is the creation of a good training dataset, one that is representative of the system and allows the network to generalize appropriately, avoiding underfitting or overfitting. A green screen was used as a determinant to create datasets quickly and to easily extract labels by performing a binarization of the matte versions. In this manner, we collected a lot of unique foreground shapes and accurate segmentation masks. Images of the dataset were built by blending the foreground shapes with background images, which were extracted from videos recorded in the area where the camera could move. In this way, the neural network takes advantage of the knowledge of that area. Various measures were used to avoid overfitting problems, such as balancing the final layer weights and using many different and unique foreground shapes. Since no such good initial results were obtained, we realized some issues in the dataset creation and addressed them. In particular, we conducted a study about high and low frequencies in images and applied a nonlinear filter to smooth the edges of the shape to make our dataset as uniform and real as possible. Subsequently, we tested our approach and carried out quantitative and qualitative assessment of our network, comparing it with some interesting approaches. Ultimately, our method is competitive and shows a good level of precision of the segmentation mask.

Furthermore, our network output can be used as an initial step to solve image matting problems: an accurate trimap can be generated quickly, starting from an RGB photo with one or more people in the foreground. It must be taken in the chosen area for network training, such as a room or an outdoor location.

Table 11 Comparison between our approach and Pix2Pix and Yolact networks trained with our dataset

Neural Network	Architecture/Backbone	Inference Time (s)
Our	encoder–decoder	0.06
Pix2Pix	U-Net	0.05
Yolact	ResNet101-FPN	0.02

The third column presents the average inference time per image in seconds on the Titan Xp GPU card

Table 12 The metrics calculated on a representative set of real photos captured in the indoor environment

	Global Acc.	Mean Acc.	Mean IoU	Mean BF Score
Our Network	0.99289	0.98649	0.96947	**0.97509**
Photoshop	**0.99416**	**0.98982**	**0.97484**	0.94339
Pix2Pix	0.97297	0.96365	0.90942	0.84006
Yolact–Our Train	0.9936	0.98349	0.9722	0.96061
Yolact-500	0.9623	0.86231	0.83671	0.88406

The highest values for each metric are highlighted in bold. Our network achieved the best value in the case of the average BF score, while the difference between our other metrics values and the best ones is less or equal to 0.5%

8.1 Limitations and future works

Despite the numerous advantages, there are some questions concerning the limitations of our approach. The main problem is due to its use of deep learning, which requires a large amount of data to obtain good results. This data must be as precise as possible to allow the neural network to obtain a good level of generalization for certain behavior. Another problem concerns the precision of the segmentation when the foreground color is chromatically similar to the background color, which makes it difficult to detect the edges of a shape. Other limitations are due to reflections within a scene (for example, glasses, windows) and motion blur (partially solvable by recording slow-motion video, from 120 fps) because the neural network fails to classify pixels in such areas. We intend to continue the development of our approach by increasing the dataset to obtain more accurate classification, by performing the segmentation on high-resolution images [37, 65] and by solving the limitations exposed in this section. Furthermore, our work is related to the chosen background area. A possible future development is to generalize the dataset in such a way that the network is independent of the type of background. At a later stage, this work could be used as an initial phase to solve the problem of human actions recognition in video surveillance systems. Finally, it could be interesting to bring our work on embedded systems, making it faster and

Table 13 The metrics computed for each class concerning a representative set of real indoor photos

|		Acc.	IoU	Mean BF Score
Our Network	Foreground	0.97784	0.94708	**0.96487**
	Background	0.99514	0.99185	0.98531
Photoshop	Foreground	**0.98395**	**0.95637**	0.91511
	Background	0.99569	0.9933	0.97167
Pix2Pix	Foreground	0.94985	0.85079	0.78263
	Background	0.97745	0.96805	0.89749
Yolact–Our Train	Foreground	0.96982	0.95173	0.94057
	Background	0.99716	0.99267	0.98066
Yolact-500	Foreground	0.72714	0.71507	0.91648
	Background	0.99747	0.95836	0.89818

The highest values for the foreground predictions are highlighted in bold
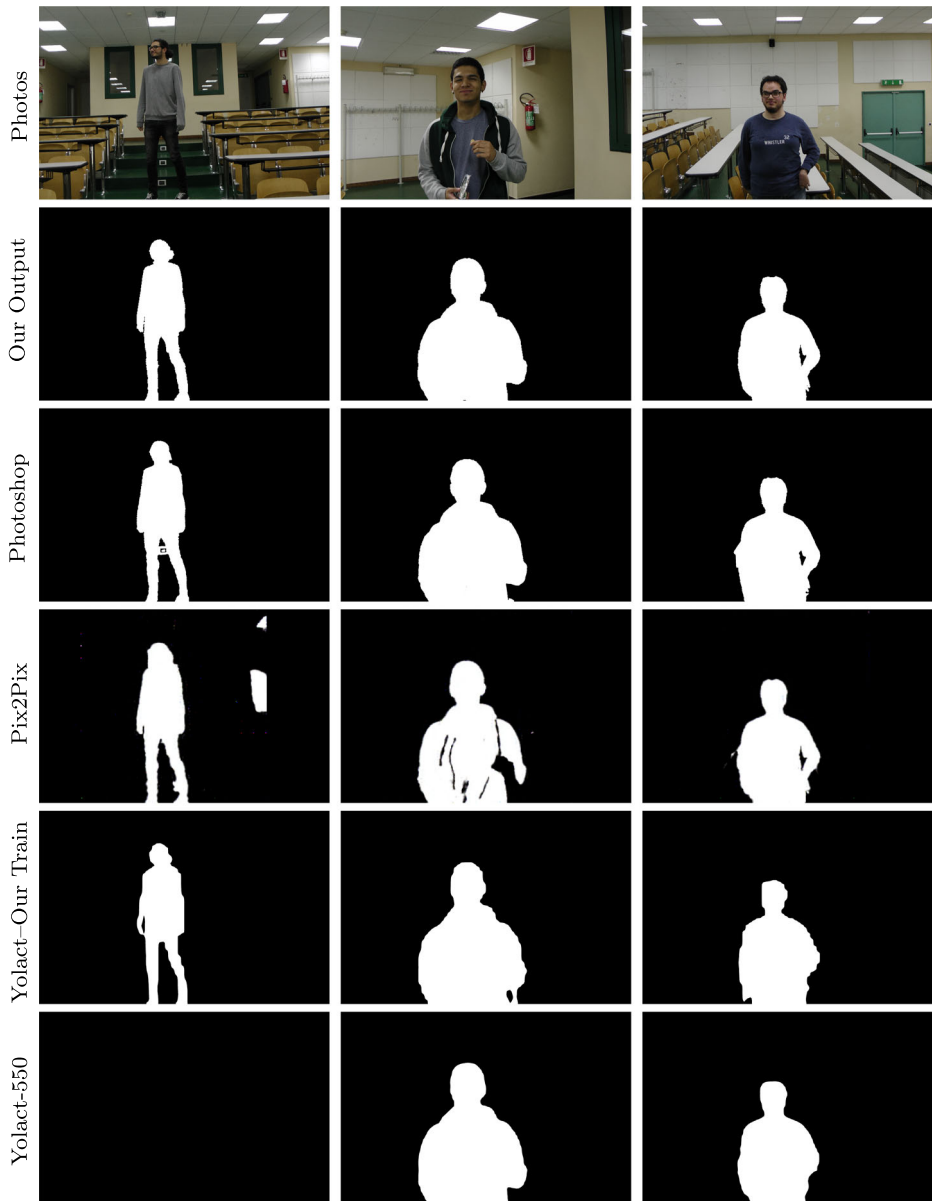
Fig. 10 Results on real photos taken directly with a camera. Both the camera and people can move in the chosen indoor area. The output masks obtained with our approach (second row) and the comparison algorithms are shown

more efficient, for example, by exploiting the potential shown by the results of recent neural network architectures.
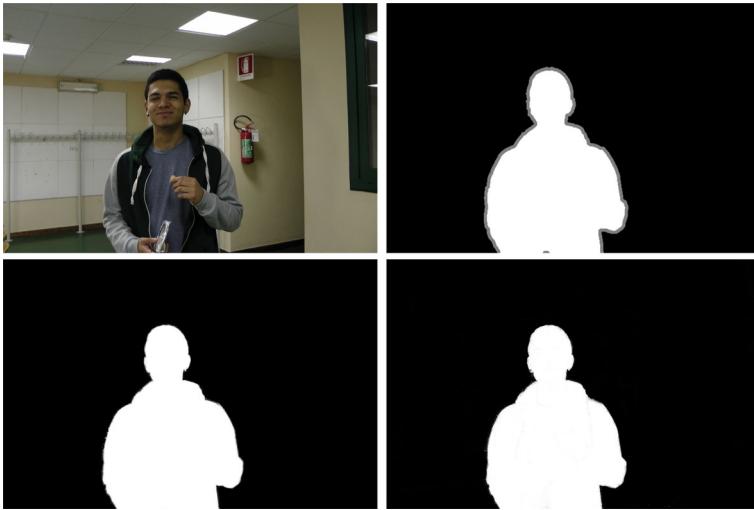
Fig. 11 The trimap generated through our network output. The top left panel shows. The network output was used to obtain the trimap (top right panel), in which the unknown region was obtained through morphological operations. It can be used as input of well-known matting algorithms, such as KNN and KL-divergence, obtaining the last two images

References

1. Abbas Q, Ibrahim ME, Jaffar MA (2018) Video scene analysis: an overview and challenges on deep learning algorithms. Multimed Tools Appl 77(16):20415–20453
2. Anthimopoulos M, Christodoulidis S, Ebner L, Geiser T, Christe A, Mougiakakou S (2018) Semantic segmentation of pathological lung tissue with dilated fully convolutional networks. IEEE J Biomed Health Inform 23(2):714–722
3. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
4. Banterle F, Corsini M, Cignoni P, Scopigno R (2012) A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. Comput Graph Forum 31(1):19–32
5. Batenburg KJ, Sijbers J (2009) Optimal threshold selection for tomogram segmentation by projection distance minimization. IEEE Trans Med Imaging 28(5):676–686
6. Bhole C, Pal C (2016) Fully automatic person segmentation in unconstrained video using spatio-temporal conditional random fields. Image Vis Comput 51:58–68
7. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer http://research.microsoft.com/en-us/um/people/cmbishop/prml/

8. Bolya D, Zhou C, Xiao F, Lee YJ (2019) Yolact: real-time instance segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 9157–9166

9. Capece N, Banterle F, Cignoni P, Ganovelli F, Scopigno R, Erra U (2019) Deepflash: turning a flash selfie into a studio portrait. Signal Process: Image Commun 77:28–39

10. Chen Q, Li D, Tang CK (2013) Knn matting. IEEE Trans Pattern Anal Mach Intell 35(9):2175–2188

11. Chen X, Zou D, Zhiying Zhou S, Zhao Q, Tan P (2013) Image matting with local and nonlocal smooth priors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1902–1907

12. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv:1412.7062

13. Dosovitskiy A, Springenberg JT, Riedmiller M, Brox T (2014) Discriminative unsupervised feature learning with convolutional neural networks. In: Proceedings of the 27th international conference on neural information processing systems, vol 1. NIPS'14. MIT Press, Cambridge, pp 766–774. http://dl.acm.org/citation.cfm?id=2968826.2968912

14. Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans Graph 21(3):257–266

15. Ess A, Mueller T, Grabner H, Van Gool LJ (2009) Segmentation-based urban traffic scene understanding. In: BMVC. Citeseer, vol 1, p 2

16. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338

17. Fernández-Caballero A, Castillo JC, Serrano-Cuerda J, Maldonado-Bascón S (2011) Real-time human segmentation in infrared videos. Expert Syst Appl 38(3):2577–2584

18. Ge F, Wang S, Liu T (2007) New benchmark for image segmentation evaluation. J Electron Imaging 16(3):033011

19. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Gordon G, Dunson D, Dudík M (eds) Proceedings of the fourteenth international conference on artificial intelligence and statistics, proceedings of machine learning research, vol 15. PMLR, Fort Lauderdale, pp 315–323

20. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT Press, Cambridge

21. Gruosso M, Capece N, Erra U, Lopardo N (2019) A deep learning approach for the motion picture content rating. In: 2019 10th IEEE international conference on cognitive infocommunications (CogInfoCom). IEEE, pp 137–142

22. Haritaoglu I, Harwood D, Davis LS (2000) W4: real-time surveillance of people and their activities. IEEE Trans Pattern Anal Mach Intell 22:809–830

23. He K, Rhemann C, Rother C, Tang X, Sun J (2011) A global sampling method for alpha matting. In: CVPR 2011, pp 2049–2056

24. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the 2015 IEEE international conference on computer vision (ICCV), ICCV '15. IEEE Computer Society, Washington, pp 1026–1034

25. Hernández A, Reyes M, Escalera S, Radeva P (2010) Spatio-temporal grabcut human segmentation for face and pose recovery. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops. IEEE, pp 33–40

26. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167

27. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134

28. Jiang F, Grigorev A, Rho S, Tian Z, Fu Y, Jifara W, Adil K, Liu S (2018) Medical image semantic segmentation based on deep learning. Neural Comput Appl 29(5):1257–1265

29. Karacan L, Erdem A, Erdem E (2015) Image matting with kl-divergence based sparse sampling. In: Proceedings of the IEEE international conference on computer vision, pp 424–432

30. Kemker R, Salvaggio C, Kanan C (2018) Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. ISPRS J Photogramm Remote Sens 145:60–77

31. Kenney J, Buckley T, Brock O (2009) Interactive segmentation for manipulation in unstructured environments. In: IEEE international conference on robotics and automation, 2009. ICRA'09. IEEE, pp 1377–1382

32. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

33. Lateef F, Ruichek Y (2019) Survey on semantic segmentation using deep learning techniques. Neurocomputing 338:321–348

34. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

35. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755
36. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–3440
37. Maggiori E, Tarabalka Y, Charpiat G, Alliez P (2017) High-resolution image classification with convolutional networks. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS), pp 5157–5160
38. Migniot C, Bertolino P, Chassery JM (2011) Automatic people segmentation with a template-driven graph cut. In: 2011 18th IEEE international conference on image processing. IEEE, pp 3149–3152
39. Morar A, Moldoveanu F, Gröller E (2012) Image segmentation based on active contours without edges. In: 2012 IEEE 8th international conference on intelligent computer communication and processing. IEEE, pp 213–220
40. Nam Y, Rho S, Park JH (2012) Intelligent video surveillance system: 3-tier context-aware surveillance system with metadata. Multimed Tools Appl 57(2):315–334
41. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 1520–1528
42. Novikov AA, Lenis D, Major D, Hladůvka J, Wimmer M, Bühler K (2018) Fully convolutional architectures for multiclass segmentation in chest radiographs. IEEE Trans Med Imaging 37(8): 1865–1876
43. Pal NR, Pal SK (1993) A review on image segmentation techniques. Pattern Recognit 26(9):1277–1294
44. Rhemann C, Rother C, Wang J, Gelautz M, Kohli P, Rott P (2009) A perceptually motivated online benchmark for image matting. In: IEEE conference on computer vision and pattern recognition, 2009. CVPR 2009. IEEE, pp 1826–1833
45. Rosenblatt F (1961) Principles of neurodynamics. Perceptrons and the theory of brain mechanisms. Tech. rep., Cornell Aeronautical Lab Inc, Buffalo
46. Sasaki Y et al (2007) The truth of the f-measure. Teach Tutor mater 1(5):1–5
47. Sengupta S, Jayaram V, Curless B, Seitz S, Kemelmacher-Shlizerman I (2020) Background matting: The world is your green screen. arXiv:2004.00626
48. Shen X, Hertzmann A, Jia J, Paris S, Price B, Shechtman E, Sachs I (2016) Automatic portrait segmentation for image stylization. In: Proceedings of the 37th annual conference of the European association for computer graphics, EG '16. Eurographics Association, Goslar. DEU, pp 93–102
49. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
50. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. CoRR arXiv:1409.1556
51. Song C, Huang Y, Wang Z, Wang L (2015) 1000fps human segmentation with deep convolutional neural networks. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR). IEEE, pp 474–478
52. Sreenu G, Durai MS (2019) Intelligent video surveillance: a review through deep learning techniques for crowd analysis. J Big Data 6(1):48
53. Stehman SV (1997) Selecting and interpreting measures of thematic classification accuracy. Remote Sens Environ 62(1):77–89
54. Tesema FB, Wu H, Zhu W (2018) Human segmentation with deep contour-aware network. In: Proceedings of the 2018 international conference on computing and artificial intelligence. ACM, pp 98–103
55. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Sixth international conference on computer vision (IEEE Cat. No.98CH36271), pp 839–846
56. Tseng YH, Jan SS (2018) Combination of computer vision detection and segmentation for autonomous driving. In: 2018 IEEE/ION position, location and navigation symposium (PLANS). IEEE, pp 1047–1052
57. Vineet V, Warrell J, Ladicky L, Torr PH (2011) Human instance segmentation from video using detector-based conditional random fields. In: BMVC, vol 2, pp 12–15
58. Xu N, Price B, Cohen S, Huang T (2017) Deep image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2970–2979
59. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision—ECCV 2014. Springer International Publishing, Cham, pp 818–833
60. Zeiler MD, Krishnan D, Taylor GW, Fergus R (2010) Deconvolutional networks. In: 2010 IEEE computer society conference on computer vision and pattern recognition, pp 2528–2535

61. Zhang SH, Li R, Dong X, Rosin P, Cai Z, Han X, Yang D, Huang H, Hu SM (2019) Pose2seg: detection free human instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 889–898

62. Zhao T, Nevatia R (2002) Stochastic human segmentation from a static camera. In: Workshop on motion and video computing, 2002. Proceedings. IEEE, pp 9–14

63. Zhao T, Nevatia R (2003) Bayesian human segmentation in crowded situations. In: 2003 IEEE computer society conference on computer vision and pattern recognition, 2003. Proceedings, vol 2. IEEE, pp II–459

64. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2881–2890

65. Zhao W, Du S, Emery WJ (2017) Object-based convolutional neural network for high-resolution imagery classification. IEEE J Sel Top Appl Earth Obs Remote Sens 10(7):3386–3396

66. Zhou YT, Chellappa R (1988) Computation of optical flow using a neural network. In: IEEE 1988 international conference on neural networks, vol 2, pp 71–78