Automatic Generation of Planar Marionettes from Frontal Images

Elad Richardson and Gil Ben-Shachar
Supervised by Anastasia Dubrovina and Aaron Weltzer

Abstract

We propose a framework for creating planar puppets from frontal images of subjects captured by commodity cameras. A deep convolutional neural network is first used to localize body joints. Reverse ensembling aggregates the responses of the model on multiple affine deformations of the input. Joint locations are further refined using facial location and skin tone cues. We exploit the skeletal pose graph to create "auto-scribbles": automatically generated foreground/background scribble masks that can be used with a wide range of segmentation algorithms to directly extract the subject's body from the background. Simple segmentation aware cropping produces individual body part crops which can be used to generate a planar marionette for repositioning and animation. The proposed joint detection pipeline compares favorably with the state-of-the-art and together with the auto-scribbles enables a fully automatic segmentation method for marionette generation.

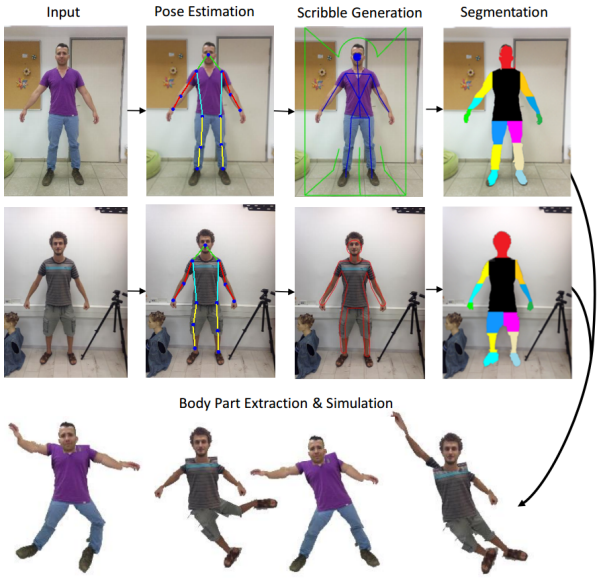


Figure 1: Visualization of the different phases of the proposed method, using either distance-based or contour-based segmentation. The last row illustrates reposing of the generated marionettes.

1. Introduction

The wide availability of modern devices with cameras has resulted in a dramatic increase in the number of pictures captured by everyday users. Simple image processing methods have thus become a common tool for regular users to improve the style and quality of their images. However, tools that allow more complicated, interactive image manipulation have been not been widely adopted by the same audience. The computational difficulties associated with such functionality result in poor usability. This is caused by lengthy and complex user experiences where multiple manual steps are required to facilitate the algorithmic pipelines. One specific functionality which is of much interest and that suffers from usability problems is full body segmentation. In this scenario one wishes to extract the body image of a person facing a camera from an image and segment it into its constituent body parts. In this note we propose a method for fully automating the body segmentation process, thus enabling a wide variety of consumer and security applications and removing the friction caused by manual input.

General image segmentation requires some sort of user input and this is often entered in the form of rough image scribbles over the object of interest. The scribble masks are then used as seeds that guide the segmentation of the object from the background [1]. The manual nature of this input process makes this approach inappropriate for general automated segmentation scenarios. Another family of segmentation algorithms take a given initial curve and uses contour evolution to move it towards the boundaries of the object [4]. The curve interior then represents the desired object however the result strongly depends on the initial curve.

For the more specific problem of frontal image body segmentation we can encode our prior knowledge of the expected image content and rough subject pose. We do this by automatically generating the required inputs for a family of segmentation algorithms using what we call "auto-scribbles" or "auto-contours". The key step in the generation of auto-scribbles is human joint detection and pose estimation. For this we need to extract from the input image

the 2D body joint locations based on a predefined human skeletal pose model. This then yields useful information regarding the object of interest which can be directly utilized for generating the desired scribbles.

An important consideration is that scribble-based segmentation algorithms tend to assume that the given input is sparse, yet accurately covers the desired foreground and background pixels without false labellings. As a result, we require precise pose estimation when producing valid input for segmentation at the next stage of our proposed pipeline. Here we note that recent successful machine learning approaches have employed either Random Forests [6] or deep Convolutional Neural Networks [26, 27, 28] to produce state-of-the-art joint localization on different public datasets. We specifically choose to build upon the neural network architecture of [26] because of it's flexibility. Crucially we show that reverse ensembling of the results on small affine distortions of the input image can significantly improve the joint localization accuracy. In addition to this we fine-tune the final detections using human body part color cues to ensure that joint detections in fact rest on actual body pixels and not on background pixels.

Accurate auto-scribbles can be derived from the extracted joint locations using our prior knowledge about the expected human shape. A foreground scribble can be generated as a skeleton-like graph connecting the detected joints. Carefully choosing those connections should result in a network of lines that provides coverage of most internal regions of the object without intersecting it's boundaries. Similarly, simple heuristics can be used to introduce a background scribble that remains sufficiently close to the object of interest without actually overlapping with it. For contour-based algorithms, we create a human-shape like auto-contour based on the detected joint locations in the image plane. Using such an approach an initial contour can breach the boundaries of the object without affecting the final result. The resulting output scribbles or contours feed into the segmentation algorithm yielding an automatic segmentation procedure.

By solving both the joint localization and segmentation problems we can exploit the results to introduce a body part aware cropping technique. This then further enables separating the observed person into their constituent components such as arms, legs, torso, head, and then using images with a transparency channel active in the region of the background segmentation mask. We demonstrate the generation and simulation of a re-posable and physically actuated planar marionette using our pipeline.

2. Related Efforts

To the best of our knowledge the proposed algorithmic scheme is the first to use pose estimation and image segmentation to explicitly crop body elements for the problem of image based subject manipulation. Direct comparison to related works is therefore somewhat challenging. However, individually, pose estimation and image segmentation have a rich history in the field of computer vision. For pose, the Pictorial Structures (PS) model was introduced by Fishler and Elschlager [12] in the 70s. It was later exploited by Felzenszwalb and Huttenlocher in [10] and then culminated in the now classic Deformable Part Models [9] for pose modeling. Similar to the PS method, Yang and Ramanan [31] introduced a flexible mixture model learned using Support Vector Machines (SVM), while Johnson and Everingham [17], suggested to integrate a set of different classifiers thereby providing a more robust approach to part detection. Following the improved ability to train deep neural networks using GPUs [19] several methods were introduced to exploit CNNs for human pose estimation. Work by Tompson *et al.* [26, 27], successfully used a multi-scale CNN for rough initial estimation of joint locations and refined the final pose using a graphical model. Similarly the DeepPose method, introduced by Toshev and Szegedy [28], uses a CNN to perform an incremental coarse-to-fine localization of joints. Interestingly, recent work by [14] has shown that DPMs are in fact neural networks, further justifying the transition to neural network based approaches.

Although pose based methods provide joint localization they do not directly provide pixel level labelling of body components. There is however an intrinsic relationship between object part boundaries and the joints which define their extent. Without pose knowledge, extracting bounded areas of general objects with pixel labels is typically performed by general image segmentation methods. These commonly use sparse user-provided scribbles together with various image statistics in order to initialize their segmentation pipelines. Classic graph cuts based techniques try to produce a foreground/background segmentation by solving a min-cut/max-flow problem over the image, as done by the GrabCut family of algorithms [24, 25]. Somewhat related to the graph based approach are those that exploit weighted geodesic distances in order to produce segmentation maps [1]. While geodesic distances are usually approximated using Dijkstra's algorithm on the graph of image pixels, other methods utilize alternative image based graphs in order to obtain more refined results, such as the level set tree approach of Dubrovina *et al.* [7]. Another classic approach in computer vision for performing image segmentation is by directly locking on to an object's contour such as in the implicit curve evolution approach of Castelle *et al.* [4]. Interestingly, although this method uses an implicit and topologically closed representation of an object contour, we can still use our auto-scribble based initialization as we will show in Section 3.2.1. In contrast to general segmentation approaches, semantic segmentation methods learn the visual characteristics of a given set of object classes. CNNs have

recently been applied in this domain and shown promising results. The scene-labeling method by Pinheiro and Collobert [23] and the the Fully Convolutional Networks (FCN) proposed by Long *et al.* [20] are two such examples. A notable extension of [20] is the end-to-end training of a FCN with a Conditional Random Field (CRF) introduced by Zheng *et al.* [32]. This method uses global and local cues to produce more refined segmentations compared to previous efforts.

Similar to our work, the connection between pose and segmentation has been explored by other researchers. The model proposed by Wang and Koller [29] optimizes both pose and pixel-wise segmentation with a relaxed dual decomposition. They solve the two problems together by using cues from one solution to improve the other. However their results produce qualitatively jagged edges, use a computationally expensive message passing scheme and does not aim to perform part specific segmentation. Ferrari *et al.* [11] suggest an approach for upper body pose estimation that uses rough segmentation, based on the head location, in order localize the area of search for the other joints. Although their method produces rough part outlines, it does not result in part specific segmentations. similarly, the work of Kohli *et al.* [18] suggests using a CRF based approach with a "stickman" prior to produce better segmentation results but does not provide part level labels. Unlike these methods the PaperDoll framework of Yamaguchi *et al.* [30] does provide explicit part labels but is not appropriate for our problem because it is specifically aimed at recognizing clothing items, not body parts. It also depends on a nearest neighbour search with a high computational and storage penalty, something which our approach does not suffer from.

3. The Proposed Method

The algorithm consists of two main parts; in the first we apply a human pose estimation algorithm to the image, generating the human pose from which we automatically derive the so called auto-scribbles. Those scribbles are then used in the second phase as priors for our segmentation algorithm. We start by describing the proposed human pose estimation method.

3.1. Human Pose Estimation

Our baseline approach is derived from the method suggested by Tompson *et al.* [27] as it has been shown to produce state-of-the-art results on challenging datasets. The CNN network takes an input image and returns a set of heat-maps, one map per each joint, representing the probability of the joint to reside at different locations in the image. These heat-maps are then used to derive the pose estimation, as shown in figure 2.

Figure 2: Pose estimation results of our method.

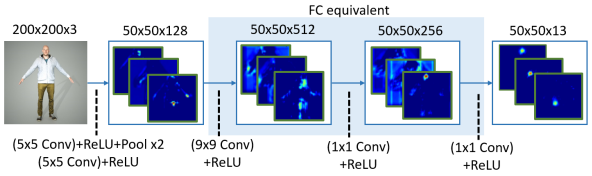


Figure 3: The neural network architecture, depicting the feature maps during the process. The layers are described between the matching feature maps.

3.1.1 The Network Architecture

A general solution would be to apply a sliding window part detector over an input image resulting in a class label probability vector for each pixel. Concatenating all the pixels results in a multichannel probability map with the i^{th} channel representing the localization probability or heat map for the i^{th} joint. Each map therefore encodes the likelihood that a joint will be localized at a particular location as can be seen in Figure 4.

Because we use a CNN the network convolutions directly provide this sliding window framework. When an input image is fed into the network, the convolutional filters are applied to each patch locally, resulting in dense feature maps describing local features of the patches. Those feature maps are then passed through another set of convolutional filters emulating a fully connected network. This results in the neural network architecture presented in Figure 3, which is fed with an input image as a whole, and returns an heat-map over the image for each joint. Further details of the architecture can be seen in the supplementary material to this paper.

One can see that the disadvantage of this method, compared to a sliding-window approach, is the significant loss of resolution during the pooling steps, though the provided resolution was still sufficient for our needs.

3.1.2 Extracting the Joints

The model described above takes an image and returns a set of heat-maps matching each joint. If we assumed that

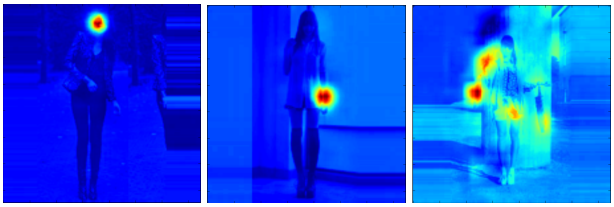
predicted heat-maps were accurate then one could simply take the maximal value from each map and denote it as the matching joint location. However, the locations provided by the CNN can be very unstable. In other words small variations in the input can lead to a large variation in the predicted location of the joint. We now introduce our approach for processing the heat-maps and localizing the joints.

3.1.3 Reverse Ensembling

In order to overcome the instability of the network we propose a form of model ensembling which we call reverse ensembling. In typical ensemble methods multiple models are trained and applied to the same input image and the results are aggregated. However here we reverse this logic and apply the same trained model to multiple deformed variations of the same image and then reverse the deformations on the predicted outputs and aggregate the reversed results. Therefore, instead of relying solely on the heat-maps generated from the single input image, a set of transformations is applied on the input image to create an augmented input set. Each transformed image is then fed into the network, and an inverse transformation is applied on the localization results. Similar to regular ensembling this method gives us a set of "different opinions" regarding the whereabouts of the joints, which can be then combined to get a statistically more stable result. Apart from the significant performance improvement we obtain using this method, we also note that the added computational complexity can be easily offset by running each prediction in parallel.

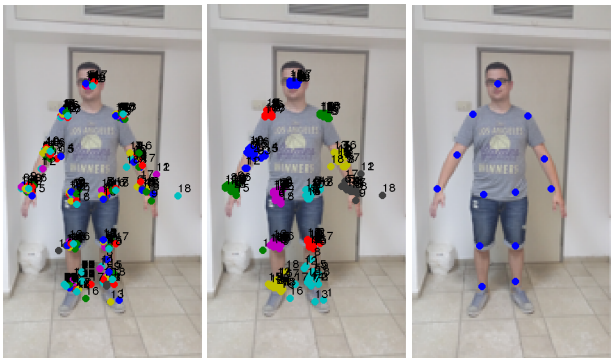
A straightforward way to aggregate the results is to sum up all the heat-maps, resulting in a single final map, where the maximal value represents the pixel that has the maximum sum. However we improve on this simple linear combination by defining a confidence value for each heat-map. By examining our predicted heat-maps we found that good results tend to look like Gaussian and are concentrated around the joint location, as defined in our training set. However, when the network has a low confidence in the joint location, a more scattered and sparse heat-map over the image is obtained as shown in Figure 4.

The pixel with the highest probability is extracted from the joint specific heat-map, and a new heat-map is generated with a Gaussian centered around the extracted pixel. The joint prediction confidence is then defined by the Mean Square Error (MSE) between these two heat maps, normalized by sum of energy of the two maps. High MSE values yield low confidence, and vice versa. For the reverse enembling these confidence values are used to calculate a confidence weighted-sum of the heat-maps thus reducing the influence of incorrect estimations on the final result. An example of the stabilizing effect of reverse ensembling can be seen in Figure 5.



(a) (b) (c)

Figure 4: Heat-maps results with different confidence values. (a,b) are results with high confidence, while (c) is one with low confidence.



(a) (b) (c)

Figure 5: The process of ensembling. In (a) we can see the different results colored by the originating network, while in (b) the results are colored by the matching joint. Though the results are centered about the right joint, there are notable deviations. Figure (c) shows the ensembled result.

3.1.4 Color-based Tuning

One of the more difficult set of joints to find with sufficient accuracy are the wrist joints; as they are often thin and even small deviations can place a prediction outside the boundaries of the body. In order to improve the localization accuracy of the wrist joint we define a wrist specific correction mechanism. To do so we note that, unlike the wrist, the head is usually detected with high accuracy, as it covers a relatively large area and includes distinctive features. Given the head location we can infer the color of the subject's face and therefore general body skin tone which will also apply to the wrist. This information can be utilized for refining the joint location.

To obtain the skin color profile we predefine a set of points about the predicted head location with the sampling radius automatically set to be proportional to the distance between the detected shoulder locations. The gathered samples are then used to generate a skin likelihood map. We then sum this with the wrist localization heat maps and recompute the predicted wirst locations. This correction

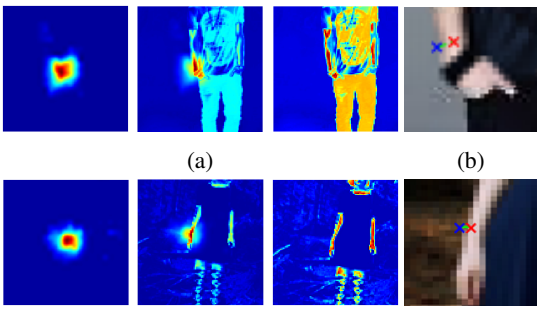


Figure 6: The process of tuning. (a,c) show the original joint heat-map on the left column, the pdf heat-map on the right column, and the combined heat-map in the middle column. While (b,d) show in red the extracted maximum after tuning, compared to the the original maximum in blue.

mechanism causes detections placed off of the body to be nudged towards the base of the hand as can be seen in Figure 6.

3.2. Full Body Segmentation

We perform segmentation by using any of a family of segmentation procedures which depend on scribble based initialization. We briefly cover the general approach of these methods and then describe our auto-scribble approach.

3.2.1 Scribble Based Segmentation Algorithms

Generally, in scribble based or contour based segmentation, the algorithm is provided with a pair of input scribbles or an initial object contour. Let $\mathcal{F}$ be the foreground scribble and $\mathcal{B}$ be the background scribble. The group of pixels labeled by the label $\mathcal{L} \in \{\mathcal{F}, \mathcal{B}\}$ will be denoted as $\Omega_{\mathcal{L}}$. From those input scribbles, a foreground likelihood function can be derived for the whole image. First, a color profile for the foreground and background is obtained from $\Omega_{\mathcal{F}}$ and $\Omega_{\mathcal{B}}$, by generating color likelihood maps from the scribble pixels. We denote these likelihoods by $Pr(x|\mathcal{F})$ and $Pr(x|\mathcal{B})$, respectively. The relative likelihood for a pixel to belong to the foreground is then defined as

$$P_{\mathcal{F}}(x) = \frac{Pr(x|\mathcal{F})}{Pr(x|\mathcal{F}) + Pr(x|\mathcal{B})} \tag{1}$$

For contour based algorithms, one first defines an initial curve inside the resulting likelihood image. A contour evolution algorithm is then applied to deform the curve towards the object boundaries. We use a piece-wise constant segmentation model based on Chan and Vese [5], applied to the likelihood image $P_{\mathcal{F}}(x)$ with geodesic active contour regularization [4] and contour evolution performed using the level set framework of Osher and Sethian [22].

In a similar fashion, in distance-based algorithms [1, 7], one first defines the weighted geodesic distance between a pair of pixels

$$d(s_1, s_2) := \min_{C_{s_1,s_2}} \int_0^1 |W \cdot \dot{C}_{s_1,s_2}(p)| dp \tag{2}$$

where $C_{s_1,s_2}(p)$ is some path connecting the pixels s_1 and s_2, and the weight function is derived directly from the likelihood function such that $W = \nabla P_{\mathcal{F}}(x)$.

The weighted geodesic distances are then calculated from both scribbles to each and every pixel by applying distance calculation on the image graph [1] or on the level-set tree of the image [7, 3, 13]. The segmented object is finally defined as the group of pixels that are closer to the foreground scribble than to the background scribble based on the respective weighted geodesics distance maps.

3.2.2 Auto-scribbles

In order to eliminate the need for user input we exploit the estimated pose to generate auto-scribbles. These are heuristically defined scribbles which are problem specific. It is useful to note however that the methodology is not unique to a human form and could be applied for other articulated objects.

The extracted joint locations are first used to create a skeleton-like foreground scribble, connecting the detected joints. We predefine the scribble connections between joints in order to generate a scribble that covers most internal regions of the subject, without breaching the background. The background scribble is introduced using simple heuristics based on human proportions, resulting in a scribble that is sufficiently close to the object of interest, without actually intersecting with it. Some resulting scribbles can be seen in the first row of Figure 7. Note that a larger region is marked inside the head as it involves a relatively larger support compared to the rest of the joints. Furthermore the feet and wrist scribbles are extended in proportion to the body size in order to capture both the hands and the shoes. For the contour based algorithms, a human-like contour is defined by using the outer contour of a binary morphological dilation of the previously defined foreground scribble. Examples can be seen in the second row of Figure 7. We note that although this initial contour may in fact breech the boundaries of the object it will subsequently be corrected by the robust curve evolution.

3.2.3 Body Part Cropping

Once the auto-contour or auto-scribbles have been generated the next step is to apply the chosen segmentation algorithm. This results in a direct full body segmentation

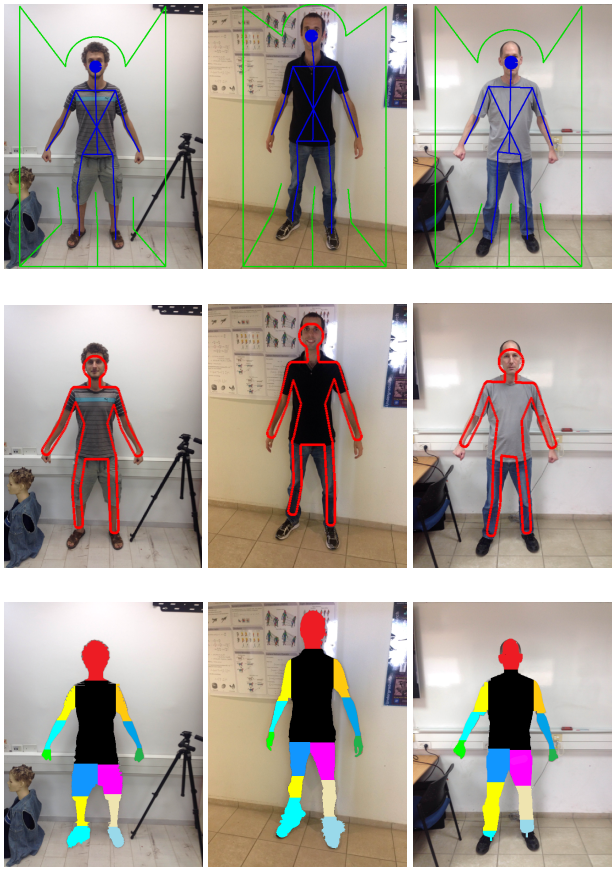


Figure 7: Automatically generated initial shape contours and extracted part label maps.

which separates the subject from the background. The subject's sub-segments are then extracted by defining cropping regions aligned with links defined between joint locations of specific limbs and body parts. Together with the background label this step provides a body part identity labeling of every pixel in the input image. An example of such a labeling and sub-segment crop shapes can be seen in the third row of Figure 7.

4. Implementation Considerations

4.0.4 Training considerations

Several datasets exist for human pose estimation. We trained our model on the FashionPose dataset [6], as it contains a set of images with relatively natural and front facing poses, as required for body parts extraction. Other datasets, like the LSP dataset [16], were too general for our needs. We first resized images to a fixed 200×200 frame a Local Contrast Normalization (LCN) filter to normalize intensity. Probability maps were created for each joint in the form of a 2D Gaussian concentrated around the ground-truth's joint location. We use Caffe [15] for our CNN implementation. To prevent overfitting the dataset was augmented by adding transformed variants of the existing images. The transformations included flipping around the horizontal axis, and rotations of the image by a small angle (± 10 degrees). One challenge we encountered was that the network tended to enter insignificant local minima. For many training configurations the network successfully learned most of the joint heatmaps, except for a random heat-map that would be zeroed for every input. Once a heat-map is zeroed, the network tends to keep it that way, making it hard to train a fully-working model. To overcome this we used AdaGrad [8] to train our initial network, resulting in a more stable learning process that produced results without any zeroed heat-maps. The network was then fine-tuned using simple Stochastic Gradient Descent (SGD).

The segmentation algorithm was implemented using OpenCV and its C++ interface, resulting in a fast and efficient algorithm. Dijkstra's algorithm was used as a distance function, due to its simplicity and low complexity. For the likelihood estimation the FIGTree algorithm was used [21].

4.0.5 Scribbles Considerations

It is important to understand that the segmentation algorithm is sensitive to mistakes in the scribbles. If a pixel is inside the foreground scribble, then the algorithm assumes it is a foreground pixel, and uses it for the distance computation. Thus, if the line breaches the object boundary, it could associate a background region with the foreground.

Thus, guiding foreground-scribble to pass through narrow regions with low probability of the joint locations could be risky. Instead of ignoring those regions, they can used for estimating the pdf, but avoided when considering the starting points for the distance calculation. The influence of small deviations in the pdf would not have a large impact on the estimated probability.

4.0.6 Partitioning the Extracted Object Body

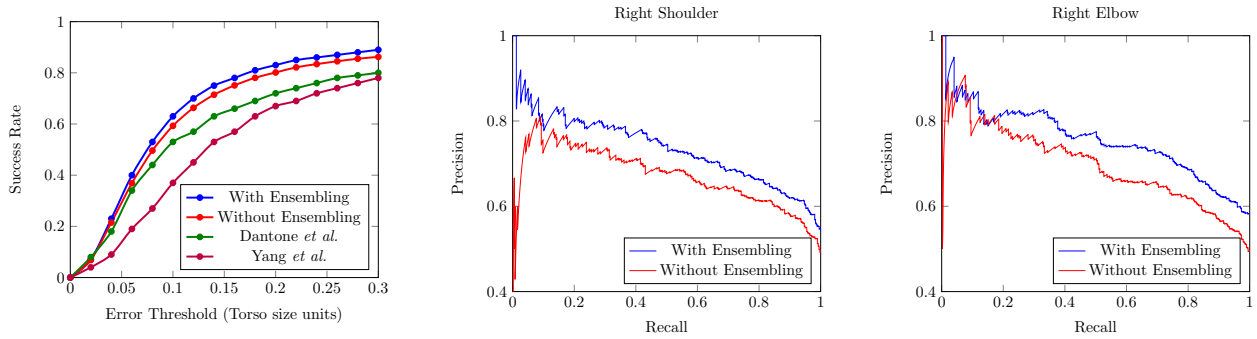
As noted in 3.2.3, the known joint locations are used to divide the segmented object into its sub-segments, thus extracting the different body parts. For exact segmentation results some more points of interest are required, such as the neckline, the intersection between the hands and the body, and the intersection between the legs. Our already found segmentation is utilized in order to find those points of interest.

4.1. Puppet Simulation

In order to demonstrate the proposed framework, a server was set-up, wrapped by a friendly user-interface, to allow users to easily upload an image and apply the above method.

	avg	Head	Shoulder	Hip	Elbow	Wrist	Knee	Ankle
With Ensembling	**62.87**	**86.01**	**80.00 78.43**	55.56 54.25	**56.47 58.30**	**43.92 46.01**	**70.20 68.10**	**60.65 59.48**
Without Ensembling	59.29	84.97	77.39 76.34	50.85 49.15	51.24 49.41	41.83 43.27	68.37 65.49	56.21 56.34
Dantone *et al.* [6]	52.02	71.35	66.84 64.13	**65.68 59.61**	43.87 40.90	33.55 28.00	55.48 53.55	47.23 46.06

Table 1: Accuracy of all the extracted joints with a 0.1 Torso units threshold. The table shows a significant improvement achieved with the ensembling technique.



(a) (b) (c)

Figure 8: Figure a shows the success rate of our Human Pose Estimation with the measure defined in 5.1. Our results are compared to Dantone *et al.* and to Yang *et al.*. Figures (b, c) Shows Precision-Recall plots with respect to the confidence measure defined in 3.1.3, comparing the accuracy of selected joints with and without ensembling.

The result is then visualized in the format of a human puppet composed of the different segmented body parts. The puppet can then be manipulated in the plane, as a ragdoll. The ragdoll itself is based on a d3.js force graph [1], with constant forces between the nodes maintaining a human form. The demo will be made publicly available on publication. A video of the procedure can be seen in the supplementary material.

5. Results

We demonstrate the results of the two main components of the suggested framework; namely, pose estimation and body segmentation.

5.1. Human Pose Estimation

The proposed joint locations estimation was tested on the FashionPose dataset, using the evaluation procedure suggested in [6]. According to this method, the euclidean distances between the ground truth and the results is normalized by the torso size. Table 1 shows our results compared to the best results of Dantone *et al.* [6] using this measure. In addition, Figure 8 shows the improvement achieved with the ensembling method compared to our basic method, over some of the joints. While Figure 9 shows the success rate of the left and center joints.

[1] http://d3js.org/

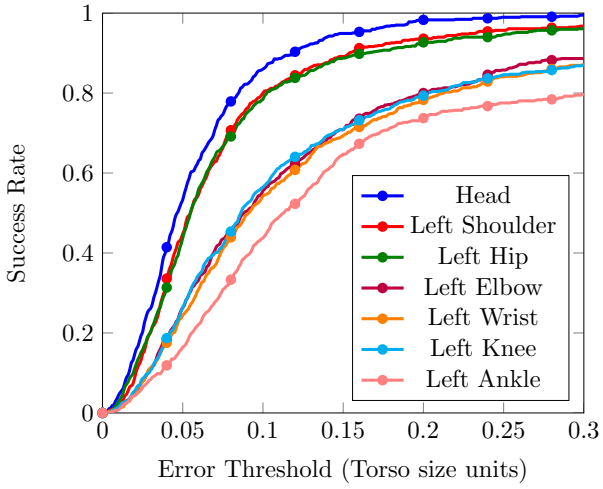


Figure 9: Accuracy plots for the head and the left side joints using the measure defined in 5.1.

5.2. Whole Body Segmentation

The body segmentation algorithm was tested using a small test set captured for our needs. The set was used to check the effectiveness of the input generation, and to compare our scribble-based implementation with the curve-

Figure 10: Comparison between different segmentation algorithm results using our input, the Semantic Segmentation algorithm [20] is included for reference. The methods are, from left to right, our curve-based implementation, our scribble-based algorithm, OneCut [25] with our generated input, and Semantic Segmentation [20].

based one. Results show that, compared to the scribble-based algorithm, the contour-based variant was less prone to take small artifacts from the background, due to the smoothness constraint. However, it does not guarantee continuity of the segmented object, as the curve might evolve into several smaller ones. Experiments also show that using the likelihood function for weights can cause the segmentation to bleed into the background, in images where a high similarity exists between the object and the background.

Our automatic input generation was also used as input for the OneCut Algorithm presented by Tang *et al.* at [25], showing that it can be easily transferred to similar algorithms. One can see that our automatic pipeline was able to produce good results on the samples shown in Figure 10. The Semantic Segmentation algorithm of Long *et al.* [20], is also presented as a state-of-the-art reference for automatic segmentation.

6. Conclusions

We proposed a framework that combines Neural Networks with model based segmentation for automatic extraction of frontal poses of human images. Using Neural Network in an indirect manner, allowed us to exploit its power while avoiding the need for creating a large dataset specific to the task. As part of the method, an ensembling technique that uses an augmented input data was introduced. The use of ensembling improved the accuracy of the estimated positions of the detected joints. More importantly, it eliminated most cases in which the estimated location of a joint was far from its real location, providing a reliable input for the automatic segmentation phase.

An important contribution is the replacement of the user input scribbles in a classical interactive segmentation algorithm by a result provided by a Neural Net. This concept can be adopted and adapted for a wide variety of segmentation tasks for specific objects. Our future efforts will focus on extending the segmentation part to handle more general human poses. This could be achieved with other model based segmentation algorithms, like the Geodesic Active Contours. In that specific case, only an initial guess about the object's boundaries should be provided by the net, while a variational principle then refines the given contour and guides it to nearby locations with high gradients. We also plan to extend the simulation and incorporate the control method suggested by Bar-Lev *et al.* in [2].

References

[1] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision*, 82(2):113–132, 2009. 1, 2, 5

[2] A. Bar-Lev, A. M. Bruckstein, and G. Elber. Virtual marionettes: A system and paradigm for real-time

3d animation. *The Visual Computer*, 21(7):488–501, 2005. 8

[3] V. Caselles, B. Coll, and J.-M. Morel. A kanizsa programme. In *Variational methods for discontinuous structures*, pages 35–55. Springer, 1996. 5

[4] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997. 1, 2, 5

[5] T. F. Chan, L. Vese, et al. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001. 5

[6] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Body parts dependent joint regressors for human pose estimation in still images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(11):2131–2143, 2014. 2, 6, 7

[7] A. Dubrovina, R. Hershkovitz, and R. Kimmel. Image editing using level set trees. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 4442–4446. IEEE, 2014. 2, 5

[8] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011. 6

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *In Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 2

[11] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *In Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008. 3

[12] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1):67–92, 1973. 2

[13] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. A quasi-linear algorithm to compute the tree of shapes of nd images. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 98–110. Springer, 2013. 5

[14] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[16] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 6

[17] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1465–1472. IEEE, 2011. 2

[18] P. Kohli, J. Rihan, M. Bray, and P. H. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, 79(3):285–298, 2008. 3

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 3, 8

[21] V. I. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. S. Davis. Automatic online tuning for fast gaussian summation. In *Advances in Neural Information Processing Systems (NIPS)*, 2008. 6

[22] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 5

[23] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013. 3

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004. 2

[25] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov. Grabcut in one cut. In *Computer Vision (ICCV), 2013*

IEEE International Conference on, pages 1769–1776. IEEE, 2013. 2, 8

[26] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. *arXiv preprint arXiv:1411.4280*, 2014. 2

[27] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1799–1807. Curran Associates, Inc., 2014. 2, 3

[28] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014. 2

[29] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2433–2440. IEEE, 2011. 3

[30] K. Yamaguchi, M. H. Kiapour, and T. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3519–3526. IEEE, 2013. 3

[31] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011. 2

[32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015. 3