Real-time Factored ConvNets: Extracting the X Factor in Human Parsing

James Charles
jjc75@cam.ac.uk
Ignas Budvytis
ib255@cam.ac.uk
Roberto Cipolla
rc10001@cam.ac.uk

Machine Intelligence Lab
Department of Engineering
University of Cambridge
Cambridge, U.K.

Abstract

We propose a real-time and lightweight multi-task style ConvNet (termed a Factored ConvNet) for human body parsing in images or video. Factored ConvNets have isolated areas which perform known sub-tasks, such as object localization or edge detection. We call this area and sub-task pair an *X* factor. Unlike multi-task ConvNets which have independent tasks, the Factored ConvNet's sub-task has direct effect on the main task outcome. In this paper we show how to isolate the *X* factor of foreground/background (f/b) subtraction from the main task of segmenting human body images into 31 different body part types. Knowledge of this *X* factor leads to a number of benefits for the Factored ConvNet: 1) Ease of network transfer to other image domains, 2) ability to personalize to humans in video and 3) easy model performance boosts. All achieved by either efficient network update or replacement of the *X* factor whilst avoiding catastrophic forgetting of previously learnt body part dependencies and structure. We show these benefits on a large dataset of images and also on YouTube videos.

1 Introduction

In many Computer Vision applications, great success has been observed when using convolutional neural networks (ConvNets) [15, 22, 34]. These networks are typically trained to solve a main task in an end-to-end manner using an appropriately chosen loss function. Unfortunately, this style of training makes it difficult to identify the sub-tasks being solved by the network prior to final inference. For instance, it is well known, at the first few layers of most ConvNets operating on images, an edge detection task is being solved and edge based filters are learnt [41]. However, deeper in the network, isolating and identifying sub-tasks of this nature becomes more difficult. This is very limiting, for example, if two ConvNets are designed to solve different tasks, it often resorts in costly training of *all* weights for both ConvNets, even though they could potentially be solving the same sub-problems. As an alternative, we propose *Factored ConvNets* as an approach for factoring out part of a network used for solving a sub-problem, such as edge detection, object localization, or background removal, and call this the *X* factor of the network. There are three main benefits to such an approach: (1) a simple plug-and-play style method to model building, where *X* factors from

one network can be plugged straight into a new model without the need for costly retraining, (2) overall network performance boosting by introduction of additional training data, not appropriate for the main task, but relevant for better training the *X* factor, and (3) small lightweight models adaptable to different domains by hot-swapping *X* factors from other ConvNets or updating them in a semi-supervised fashion [20].

Here we demonstrate the Factored ConvNet approach within the application of human body parsing were the task is pixel-wise segmentation of the human body into 31 parts. The ConvNet is designed to be both lightweight (low number of parameters) and fast, performing at 120fps in batch mode or 11fps through a webcam (512x512 input resolution), yet easily adaptable to new domains and capable of semi-supervised personalization to people in videos. The chosen *X* factor for this task is foreground/background (f/b) segmentation, i.e. a precise pixel-wise classification into person (foreground) or non-person (background). This is an appropriate sub-task to body part segmentation which is a more fine-grained segmentation of the foreground region. We show how to build and train the Factored ConvNet to correctly use knowledge supplied by the *X* factor, leading to four main contributions: (i) A real-time method for human parsing, (ii) easy model improvements by swapping out the *X* factor for better modules, (iii) a method of simple transfer of the network to new domains and (iv) a technique to personalize the network to people (and backgrounds) in input videos without catastrophic forgetting of previously learnt human body structure and part dependencies.

1.1 Related work

Multi-task learning. ConvNets designed for multi-task learning typical have a shared layer of features which acts as input for two or more independent network branches trained for different tasks [12, 19, 21, 36, 38]. This often results in improved prediction efficiency and increased accuracy but is hindered by the assumption of task independence. To address this, Misra *et al*. [26] recently proposed a method for learning how to 'stitch' two networks together to share information. However, this intertwining of networks inhibits task modularity. In [10] tasks are produced sequentially, with the next task dependent on the previous and in [3] a recurrent architecture learns how to interconnect tasks to help improve overall performance. While our work also benefits from a shared representation, we also show how to form intertask dependencies while keeping certain tasks modular and easily updated.

Modular networks. Using the first set of image encoding layers from *e.g*. VGG [34] or ResNet [15] can work well when fine-tuned for new tasks, but easily sharing network architecture and weights from sub-nets *within* a network in a plug-and-play style approach is still a difficult. Recently Fernando *et al*. [13] developed PathNet for unsupervised task sharing between networks. However, we tackle the problem (for the case of human parsing) in a supervised approach where the shared task is known and can also be substituted by non-network based models. Alternative methods [8, 27] grow networks or update them over time to learn new tasks, but we endeavour to have a lightweight and fast network (not a large universal problem solver) capable of adapting online to different domains.

Human body parsing. We tackle the problem of human parsing [14, 24, 29] which has seen recent advances, particularly due to the introduction of more training data [9, 16, 23, 39]. Of similar application is human pose estimation where recent ConvNets have shown very good performance [2, 5, 6, 28, 40] when designed to capture the dependencies between

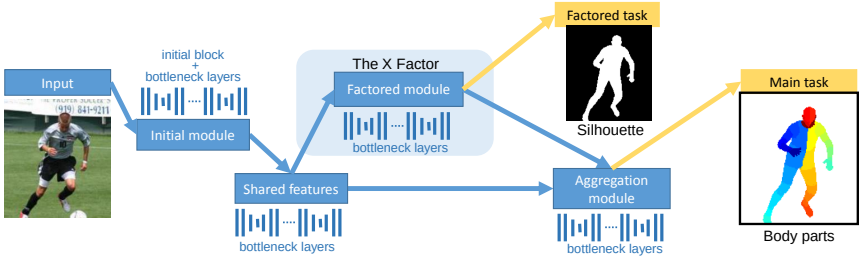


Figure 1: **The Factored ConvNet.** shown on a main task of human body parsing. Blue blocks show bottleneck and convolutional modules. Foreground/background segmentation factored out as an *X* Factor.

body parts. In our design, we capture large image context and learn global human body structure, while retaining network efficiency, by using dilated convolutions [42] and separable filters [17, 35]. Our novelty in this area is a real-time system which is capable of being personalized and also transferred to other domains. This is extremely important as human parsing training data is costly to obtain. Also, although personalized methods have been developed for human pose tracking in video [7, 31], we believe this is a first for human parsing in video.

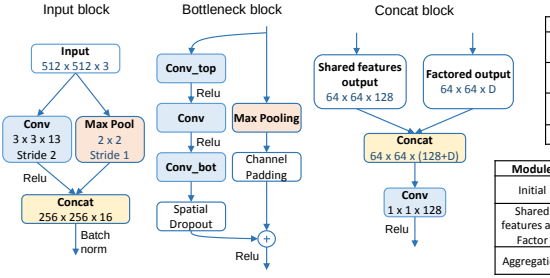
2 Network overview

For the Factored ConvNet to be efficient at run-time and also capable of being updated quickly during domain transfer or personalization, we use similar architectural choices to the ENet network [30] which is also fast and lightweight. Our network has two tasks, a main task and a chosen factored task, illustrated in Figure 1 for the main task of human parsing (body part segmentation) and sub-task of f/b segmentation.

Factored ConvNet. Our feed-forward network can be applied to both images and video and operates on each frame/image independently. The network is composed of convolutional modules (indicated as blue blocks in Figure 1). The input image is first down-sampled through an initial block and shared features are later computed for two separate tasks: (1) the factored task and (2) the main task. An aggregation module combines the outputs from the factored task with shared features prior to inference on the main task. In this manner, it is clear where in the network the sub-task (f/b segmentation) is computed, i.e. the factored module.

The *X* Factor. The factored module and it's associated task define the *X* Factor. Any factored task that is related to the main task but for which training data is more prolific or easier to obtain would be ideal for an *X* Factor. In the case of human parsing, this could be 2D human pose estimation or person detection. We later show that the performance of the main task can then be boosted by simply using additional training data when training the factored module. To enforce the factored module to reason about the *X* factor's task an associated loss function is attached.

Aggregation module. This module makes the Factored ConvNet different to traditional multi-task networks. Rather than each task being solved independently after the shared feature stage, here information coming from the factored task is merged back into the network



Bottleneck blocks

Layer	B_reg	B_asym	B_dilateX	B_down
Conv_top	1 x 1 Stride 1	1 x 1 Stride 1	1 x 1 Stride 1	2 x 2 Stride 2
Conv	3 x 3	5 x 1 and 1 x 5	3 x 3 Dilated by X	3 x 3 Stride 2
Max Pooling	No	No	No	2 x 2 Stride 1
Conv_bot	1 x 1	1 x 1	1 x 1	1 x 1

Module architectures

Module	Blocks							
Initial	Input block	B_down	B_reg	B_reg	B_reg	B_reg	B_reg	
Shared features and Factor	B_reg	B_dilate2	B_asym	B_dilate4	B_reg	B_dilate8	B_asym	B_dilate16
Aggregation	Concat block	Same blocks as shared features and factored module						

Figure 2: **ConvNet modules.** Diagrams showing the three fundamental building blocks for the ConvNet modules. Top table details the four types of bottleneck blocks used [30]. Bottom table details blocks used within modules from left to right, indicating the direction of information flow through the network.

by combining it with the information stream from shared features. For best performance, it is important for the aggregation module to 'know' when to attend to or when to ignore the information coming from each stream. For example, in some cases poor lighting can make specific body part detection difficult, but background subtraction can still be performed easily. Under such conditions the network should learn to pay more attention to information from the factored module when solving the main task (see Section 4 for how this behavior is trained). This type of reasoning is lost in traditional multi-task setups.

3 Benefits to factoring

Due to task dependencies, boosts in overall network performance can be obtained by only having to improve X factor accuracy, making it easier to transfer the network to new domains or personalize it to an input video.

Domain transfer. Compared with traditional ConvNets, factored ConvNets can be easily transfered to different domains where training data is available for the X factor but not available for the main task. Transfer is simply done by updating the factored module with additional training data, while keeping the rest of the network fixed to avoid catastrophic forgetting of previously learnt knowledge for computing the main task. The ConvNet can also be transfered in a semi-supervised approach onto a collection of images (detailed in Section 4), or if the X factor is already provided in the new domain (e.g. f/b silhouettes), one can directly inject it into the network for immediate performance boosts on the main task (see Figure 3(a)).

Personalization. Similar to Charles *et al.* [7] we personalize the Factored ConvNet through fine-tuning, however, in our case, only a portion of the network (the factored module) has to be fine-tuned with training data of the target person. Personalization can be fully automatic through semi-supervised learning, particularly if the X factor is a much easier task to the main task. In the next section we provide the details of our model under the application of human body parsing (31 body part segmentation) and describe a semi-supervised approach for personalizing the Factored ConvNet to a target person merely by using their silhouette.

4 Method details

Factored ConvNet architecture. Three fundamental blocks (illustrated in Figure 2) are employed to build the Factored ConvNet. Each of the modules (see Figure 1) is composed of different types of ENet [30] style bottleneck blocks. Exact construction is detailed in the tables shown in Figure 2. The ConvNet is trained on 512x512 RGB images and final resolution of the output segmentation is 64x64. Up-sampling is not performed here as evaluation at the low resolution is sufficient to demonstrate the benefits of factoring out sub-tasks. Batch normalization is used between all conv layers except where stated otherwise. Channel padding [30] in bottlenecks is used to match the number of feature maps prior to summing.

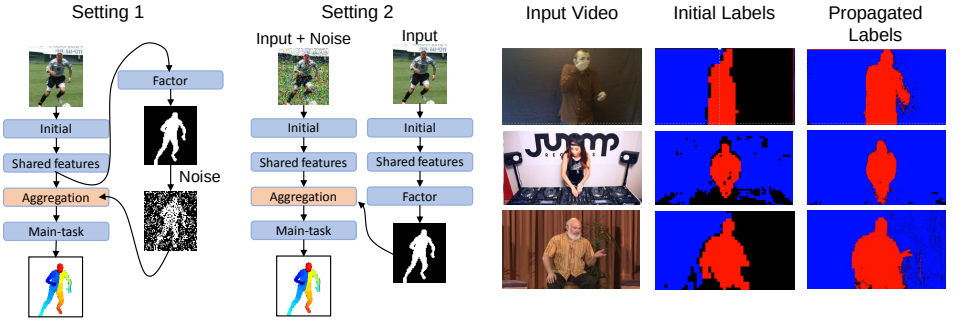
Losses. An associated prediction layer and loss is appended to the factored module. In our case a 1×1 conv layer and softmax activation predicts two classes (foreground/background) and a cross entropy loss is assigned. In a similar fashion for the main task, a prediction layer of 32 classes (body parts + background) and cross entropy loss is appended to the aggregation module. During training we found both losses weighted equally worked well.

Aggregation module. A concat block (see Figure 2) lies at the head of this module and concatenates output feature maps (128 channels) from the shared feature module with output class prediction maps (2 channels) from the factored module.

Training. The network is trained end-to-end when provided with appropriate training data. However, it is important to ensure the aggregation module learns how much attention to pay towards signals coming from either the factored or shared feature modules. This is taught by injecting noise at specific points in the network under four types of settings. Each setting gives rise to different learnt behavior. For *Setting 1*, noise is injected at only the output of the Factored module, accomplished by swapping confidence values (between f/b classes) at random locations in the prediction map (see Figure 3(a)). *Setting 2*, the initial block of the network is cloned, producing two parallel streams through the network, one stream connecting to each module (see Figure 3(a)). Gaussian noise is added to the input image of the shared feature stream, while keeping the input to the factored module clean. In *Setting 3*, no noise is injected and for *Setting 4* both types of noise from *Settings 1* and *2* are added. One setting per training batch is chosen at random. *Setting 1* ensures the network can reason about image features when classifying pixels into body parts, *Setting 2* forces the network to reason about silhouette shape and *Settings 3* and *4* provide balance. Other than the aggregation module, all weights in other modules are held fixed when training under *Settings 1 2* and *4*. The variance in Gaussian noise is also randomly chosen for each image, ranging between 10 and 120 pixel values.

Boosting network performance. main task predictions can be boosted by improving factored task performance in two ways: 1) *direct injection* of known silhouettes for the input image, which may come from a different source (e.g. a human user or another f/b model) or 2) *Factor fine-tuning*, where the factored module is fine-tuned on training images of people and their silhouettes while the weights of all other modules are fixed.

Domain transfer. If provided with target domain training data of human silhouettes, the Factored ConvNet is transferred by *factor fine-tuning*, (or even by *direct injection* if silhouettes are provided on all images in the new domain). For semi-supervised transfer, training images of human silhouettes are automatically obtained by initializing Grabcut segmentation [32] with f/b silhouette predictions from the factored module. Transfer is then accom-



(a) Training the aggregation module (b) Improvements from label propagation

Figure 3: (a) Noise injected into the Factored ConvNet in two different ways during training of the aggregation module, all other module weights are held fixed. (b) Shows the semi-supervised label propagation method [4] used to help obtain silhouettes for video personalization. Areas of high, low and close to 0.5 confidence for foreground are shown as red, blue and black respectively.

plished by *factor fine-tuning* on these new silhouettes. Note, only f/b segmentation is required for transfer, this is far easier to obtain than segmented images of people into 31 body parts, and much less error prone if being produced automatically.

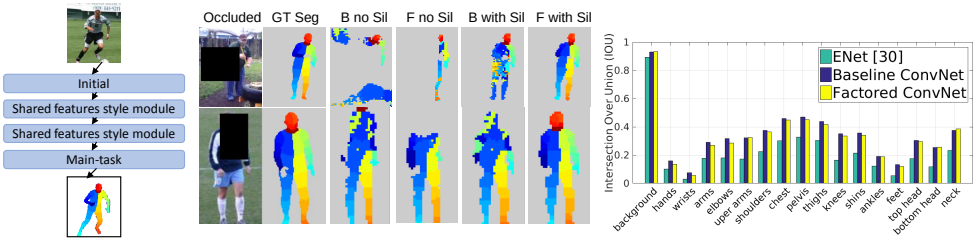
Personalization in video. Personalization is achieved by *factor fine-tuning* on automatically obtained human silhouettes from a target video. Temporal information is leveraged to obtain training silhouettes as follows: Initially predicted f/b segmentation regions in each frame (regions where the factored module has very high class prediction (>0.99)) are propagated both forward and backward in time through the video using a tree structured graphical model [4]. Unlike dense optical flow, which smooths tracking across object boundaries, this technique can produce crisp tracked silhouettes which align well with f/b edges. An added advantage is that the method has the capability to fill in any initially missed f/b regions, see Figure 3(b). As with domain transfer, semi-supervised label generation using the silhouette is much easier than obtaining labels for the main task (which would involve the complex problem of detecting and tracking body parts [7]). When fine-tuned, the module learns person and background specific features resulting in better f/b segmentation and improved human parsing by the main network.

5 Experimental evaluation

For our experimental evaluation we test the Factored ConvNet on the application of parsing an image into 31 different body parts, as introduced by Shotten *et al*. [33] (see Figure 7(b)). Two datasets are used, one for training and testing the model on still images, the other for testing personalization to video.

5.1 Datasets and training

Unite the People S31 [23]. A total of 8515 images originating from Leeds Sports Pose [18], single person tagged people from MPII Human Pose Dataset [1] and FashionPose [11]. Full-body part segmentations for 31 parts (and background) are provided by Lassner *et al*. [23] by automatic fitting of a 3D body model and part projection. Example images and body part types in different colors are shown in Figure 7(b). We randomly split this collection into 6812 images for training and 1703 for testing.



(a) Baseline ConvNet (b) Qualitative results on occlusion experiment (c) Unite the People S31 comparison

Figure 4: **Human parsing under simulated occlusion.** (a) The baseline ConvNet. (b) Human body segments from the baseline and Factored ConvNet (B and F) on example images from Unite the People S31 test set. Baseline and Factored ConvNet output with provided GT silhouettes (with Sil) and without (no Sil) are shown. Notice how the Factored ConvNet can reason well about the silhouette when forming body part classification. (c) shows per-class IOU comparison on Unite the People S31.

YouTube Pose [7]. A 50 video dataset of different people from YouTube, each with a single person in the video. For each video, 100 frames are manually labeled with human upper body pose. Five testing videos from this dataset are chosen where our model has difficulty producing good body part segmentation. All labeled frames from each of the five videos is used for testing (500 frames in total). Example frames from these videos are shown in Figure 7(a).

Freiburg Sitting People (FSP) [29]. A dataset of 201 images of six different sitting people. Each image is provided with ground truth segmentation into 14 different body parts. The train set consists of two people while a test set of four people are held out as in [29]. Models trained from Unite the People S31 are tested against this dataset by merging certain class labels (from the 31 classes) to form matching 14 body part classes.

Training. For all experiments both the baseline and factored model are trained for 260 epochs on Unite the People S31 training set. Heavy augmentation such as horizontal flips, rotation (-80 to 80 degrees), random cropping ($\pm$200 pixels off center) and scaling (0.8 to 1.2 scaling) is used. Optimization of weights is done using RmsProp [37] with a learning rate of 0.0005 and weight decay of 0.0002, training took about 1.5 days on a Titan XP GPU, no learning rate scheduling was applied.

5.2 Experiments

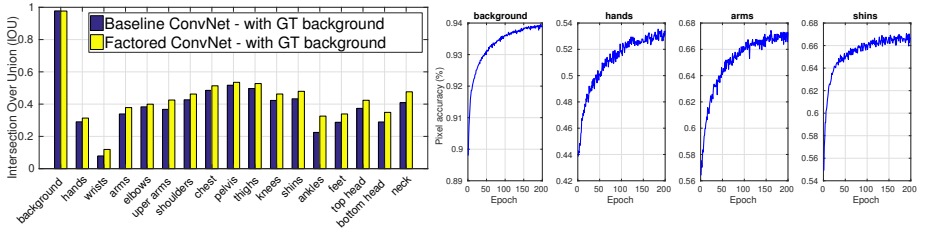
Baseline. While we also compare to the state of the art (ENet [30] and FCN [25]), a non-factored baseline ConvNet is additionally tested against. This is a very similar network to the Factored ConvNet, except without having a factored task and can be considered as a deeper version of ENet [30]. This baseline is constructed from the same number of modular components as the Factored ConvNet and has similar number of parameters.[1] For details of the baseline architecture a diagram is shown in Figure 4(a).

Evaluation metrics. Segmentation performance is scored by measuring mean per-class pixel-classification accuracy and also mean per-class intersection over union (IOU). On Unite the People, these scores are shown as averages over left-right body parts. On the YouTube pose experiment, segmentation accuracy cannot be scored directly as no ground truth labels are available for this task. Instead we measure performance indirectly by inferring body joint locations from predicted class labels and calculating their distance to ground

[1] A few extra parameters are used in the Factored ConvNet for computing the factored task prediction (1×1) convolution and also in the concat block (see Figure 2) used by the aggregation module.

Method	FCN [25]	ENet [29]	Baseline	Factored ConvNet	Transfered
Mean class accuracy (%)	59.7	28.0	68.9	69.0	**71**
Mean class IOU	0.43	0.13	0.41	0.39	**0.48**

Table 1: **Comparison on Freiburg Sitting People [30].** The Factored ConvNet performs favorably against the state-of-the-art. When transfered to this new domain (using only human silhouettes) the Factored ConvNet (Transfered) outperforms all networks.



(a) (b)

Figure 5: **Unite the People S31 experiment.** (a) Improved performance of Factored ConvNet over Baseline ConvNet when GT background silhouettes are provided by *direct injection* (b) Performance curves showing improvement over body part classes (all classes shown in supplementary material) when GT background silhouettes are used during *factor fine-tuning*.

truth body joint annotation (which is available). Centre of mass for wrist, elbow, shoulder, and combined top/bottom head segments are used to obtain 7 body joint estimates.

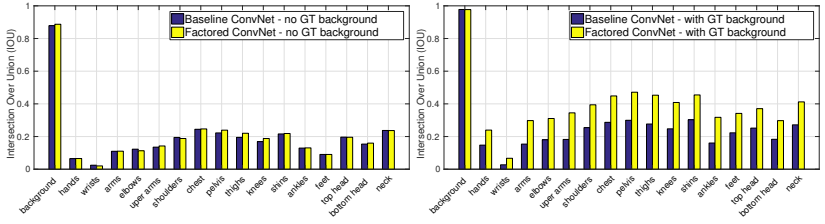
Experiment 1: State-of-the-art vs Factored. On the FPS testing data, mean pixel classification accuracy and IOU (not including background) of the Factored ConvNet, FCN [25], our implementation of ENet [30] and the baseline is shown in Table 1. Note, all networks were trained on Unite the People, other than the FCN which was fine-tuned by [29] on the train set of FSP. Performance charts showing IOU on Unite the People is shown in Figure 4(c). Both the baseline and Factored ConvNet perform similarly well with mean per class pixel accuracy of 44% and 43% respectively, with ENet falling behind at 28%. The Factored ConvNet performs favourably against all networks, and illustrates factoring is not detrimental to accuracy, yet also provides benefits which we show next.

Experiment 2: Direct injection. Here we test the degree of performance boosting capable by the Factored ConvNet on it's main task, when only improving performance of the factored module on the factored task. An upper bound is obtained by providing the network with GT silhouettes on testing images using the method of *direct injection* (see Section 4). For comparison to the baseline method, when maximizing over the baseline's softmax layer, GT testing silhouettes are used to boost it's performance by hard constraining pixels to be either one of 31 foreground classes or background. Figure 5(a) shows the benefits of factorization, with IOU of the Factored ConvNet overtaking the baseline across all body parts. This indicates the aggregation module is correctly reasoning about GT silhouette shape when making body part class prediction, rather than merely suppressing erroneous background confidences, as is only capable by a model which has no obvious insertion for sub-task knowledge. This effect is highlighted more so in the next experiment.

Experiment 3: Simulated occlusions. Situations where body part detection is difficult compared with f/b segmentation can occur in many real-life scenarios, poor lighting on the human body but uniform background, grainy video but where motion can be used for f/b

Method	BG	Hnds	Wrst	Larm	Elbw	Uarm	Shldr	Chst	Plvs	Thgh	Kne	Shn	Ankl	Ft	T-Hd	B-Hd	Nck
Orig	96.5	22.8	8.2	41.5	41.3	47.3	48.3	59.9	58.3	54.7	47.9	49.1	28.5	16.7	22.8	8.2	41.5
Tran-Init	**96.6**	24.8	9.0	43.1	42.7	49.3	49.6	61.2	57.6	54.5	46.3	46.0	26.0	13.7	24.8	9.0	43.1
Tran-Grab	93.3	**36.0**	**9.0**	**46.9**	**45.9**	**51.1**	**53.4**	**63.8**	**60.6**	**57.3**	**50.8**	**52.0**	**29.0**	**23.6**	**36.0**	**9.0**	**46.9**

Table 2: **Semi-supervised domain transfer to Unite the People S31.** Pixel classification accuracy shown.



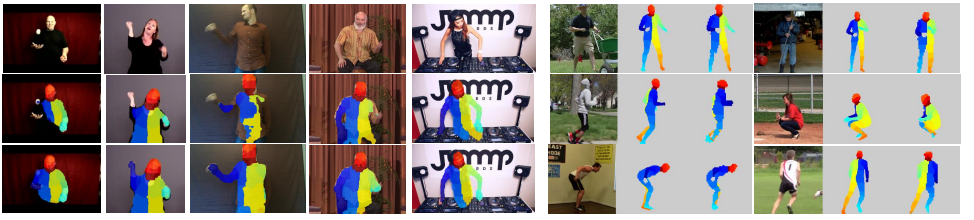
(a) (b)

Figure 6: **Simulated occlusion experiment.** Mean per-class IOU of Baseline and Factored ConvNet under simulated occlusion when (a) GT background silhouettes are **not** provided and (b) when GT background silhouettes **are** provided, notice the large improvement gains over Baseline when better background silhouettes are available.

subtraction or domains where there is more training data for the f/b subtraction task. Here we simulate difficult body part detection by occluding body parts (in test images) with a randomly placed black box (see Figure 4(b)), and ensure easy f/b segmentation by providing the baseline and Factored ConvNets with GT silhouettes (as in Experiment 2). Mean IOU over classes before GT silhouettes are provided is shown in Figure 6(a), both models perform equally. Once GT is provided, both models improve, but huge gains in IOU from the Factored ConvNet over the baseline is observed across all body part classes (see Figure 6(a)). This clearly demonstrates the ability of the aggregation module to correctly attend to the signal from the f/b segmentation task. Qualitative performance is as equally compelling, figure 4(b) shows example testing images and model outputs before and after GT silhouettes are provided. The baseline model, even when given GT silhouettes, has no chance of recovery as image features are no longer useful. On the other hand, the Factored ConvNet, which has been trained to interpret the silhouette, can successfully recover. Experiments 2 and 3, thus suggests swapping out the factored module for better trained or other types of f/b segmentation nets or methods will lead to improved performance on the main task.

Experiment 4: Supervised domain transfer. Using the method of *factor fine-tuning* the Factored ConvNet can be transfered from the training set to the testing set. Different to *direct injection* of GT silhouettes (which are perfect), the response of the factored module will alter slightly during training. Figure 5(b) shows the robustness of the aggregation module to the effects of fine-tuning, and demonstrates smooth improvements in body part segmentation as f/b segmentation becomes better. In this manner, we also transfer the Factored ConvNet to the FSP dataset using only f/b segmentation of training images. Results of this 'Transfered' model are shown in table 1 improving on the state-of-the-art and baseline.

Experiment 5: Semi-supervised domain transfer. Using the semi-supervised method detailed in Section 4, we transfer the Factored ConvNet from the train set to the test set of Unite the People. The factored module is fine-tuned for one epoch through the testing data. Table 2 shows pixel-wise classification accuracy before (Orig) and after (Tran-Grab) domain transfer. We also tested *factor fine-tuning* on initial silhouette predictions (Tran-Init) from



(a) YouTube Pose (b) Unite the People S31

Figure 7: **Qualitative examples on Youtube Pose videos and Unite the people S31 test set.** (a) Middle row is by non-personalized Factored ConvNet, bottom row is by personalized Factored ConvNet. (b) Columns 2 and 5 are ground truth, while columns 3 and 6 are by standard Factored ConvNet.

Method	Head	Wrists	Elbows	Shoulders
Non-personalized Factored ConvNet	9.1	15.4	23.5	12.8
Personalized Factored ConvNet	9.1	12.6	17.3	9.6

Table 3: **Body joint estimation on YouTube videos.** Average pixel distance from ground truth for body joint estimates (averaged over all videos and test frames, 500 frames in total), smaller is better.

the factored module on the test set (without Grabcut [32] refinement) . This also leads to some body part prediction improvements, but less so than when initializing with Grabcut.

Experiment 6: Personalization to YouTube. For each video, within a window of 301 frames around one testing frame, initial f/b segmentation labels are computed by the factored module (abitrarily, the closest test frame to the half-way point of the video is chosen). Label propagation [4] is applied to frames only within this window. The Factored ConvNet is personalized to each testing video separately using labels from only this window of 301 frames. During testing it is applied to all 100 test frames throughout the whole video. Quantitative results in table 3 show improvement from personalizing over all body joints. Qualitative results (see Figure 7(a)), shows personalization now recovers hands and arms while body part segments also fit the shape of the person better.

Model complexity. Compared to other ConvNets for segmentation, the Factored ConvNet uses very few parameters. In fact only ~0.5 million parameters are used compared to e.g. FCN [25] of ~136 million. At test time the network can process each frame live through a webcam at ~11fps (inclusive of rendering in MATLAB) or ~120fps in batchmode on a Titan XP GPU.

6 Summary and future work.

Factored ConvNets are proposed for extracting *X* factors which are localized sub-nets performing known sub-tasks, such as edge detection or background removal. Under the application of human parsing (segmentation into 31 body parts), we show how one can build a real-time Factored ConvNet with a foreground/background (f/b) segmentation *X* factor, leading to ease of domain transfer, personalization and leveraging of extra silhouette training data to improve body part classification. The *X* factor was shown to be modular and replaceable with other f/b segmentation algorithms. Adding other *X* factors to the model such as human pose estimation or even factors with relatively small training material (*e.g.* skin detection or optical flow), would likely bring benefits and we leave this for future studies.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. CVPR*, 2014.

[2] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *Proc. AFGR*, 2017.

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Proc. NIPS*, 2016.

[4] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *Proc. CVPR*, 2011.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*, 2017.

[6] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015.

[7] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proc. CVPR*, 2016.

[8] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015.

[9] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *Proc. 3DV*, 2016.

[10] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proc. CVPR*, 2016.

[11] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Body parts dependent joint regressors for human pose estimation in still images. *IEEE PAMI*, 2014.

[12] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. ICCV*, 2015.

[13] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[14] Ke Gong, Xiaodan Liang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. *arXiv preprint arXiv:1703.05446*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *pami*, 2014.

[17] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.

[18] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proc. BMVC*, 2010.

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. ICCV*, 2015.

[20] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proc. CVPR*, 2017.

[21] Iasonas Kokkinos. Ubernet: Training auniversal'convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proc. CVPR*, 2017.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[23] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proc. CVPR*, 2017.

[24] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural network. In *Proc. ICCV*, 2015.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015.

[26] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proc. CVPR*, 2016.

[27] Tom M. Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Bryan Kisiel Matthew Gardner, Jayant Krishnamurthy, and Kathryn Mazaitis Ni Lao, Thahir Mohamed, Ndapa Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, and Derry Wijaya Richard Wang, Abhinav Gupta, and Abulhair Saparov Xinlei Chen, and and Joel Welling Malcolm Greaves. Never-ending learning. 2015.

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016.

[29] Gabriel L Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. In *Proc. ICRA*, 2016.

[30] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[31] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. CVPR*, 2005.

[32] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Proc. ACM TOG*, 2004.

[33] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE PAMI*, 2013.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. CVPR*, 2016.

[36] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

[37] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

[38] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *Proc. GCPR*, 2016.

[39] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. In *Proc. CVPR*, 2017.

[40] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. CVPR*, 2016.

[41] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proc. NIPS*, 2014.

[42] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.