Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos

Tomas Pfister[1], Karen Simonyan[1], James Charles[2] and Andrew Zisserman[1]

[1]Visual Geometry Group, Department of Engineering Science, University of Oxford
[2]Computer Vision Group, School of Computing, University of Leeds

Abstract. Our objective is to efficiently and accurately estimate the upper body pose of humans in gesture videos. To this end, we build on the recent successful applications of deep convolutional neural networks (ConvNets). Our novelties are: (i) our method is the first to our knowledge to use ConvNets for estimating human pose in videos; (ii) a new network that exploits temporal information from multiple frames, leading to better performance; (iii) showing that pre-segmenting the foreground of the video improves performance; and (iv) demonstrating that even without foreground segmentations, the network learns to abstract away from the background and can estimate the pose even in the presence of a complex, varying background.
We evaluate our method on the BBC TV Signing dataset and show that our pose predictions are significantly better, and an order of magnitude faster to compute, than the state of the art [3].

1 Introduction

The goal of this work is to track the 2D human upper body pose over long gesture videos. As a case study, we experiment on a dataset of sign language gestures, which contains high variation in pose and body shape in videos over an hour in length. The foreground appearance in these videos is very varied (as shown in Fig 3), with highly varying clothing, self-occlusion, self-shadowing, motion blur due to the speed of the gesturing, and in particular a changing background (due to the person being superimposed over a moving video, as shown in Fig 2).

We build upon recent work in video upper body pose estimation. Buehler *et al.* [2] proposed a generative model capable of tracking a person's upper body continuously for hours, but required manual annotation of 64 frames per video, and was computationally expensive. Charles *et al.* later used this generative model to train a faster and more reliable pose estimator [3,4,15] using a random forest. However, their method relied on a hand-tuned foreground segmentation algorithm for preprocessing the videos, without which their method performed poorly. The segmentation method had to be manually tuned for each different type of video, and failed on certain videos with unusual body shapes, unusual absolute positions of persons in the video, or similar foreground and background colours. Further, the extensive preprocessing was computationally expensive, reducing the speed of the method to near-realtime. In our method (pictured in

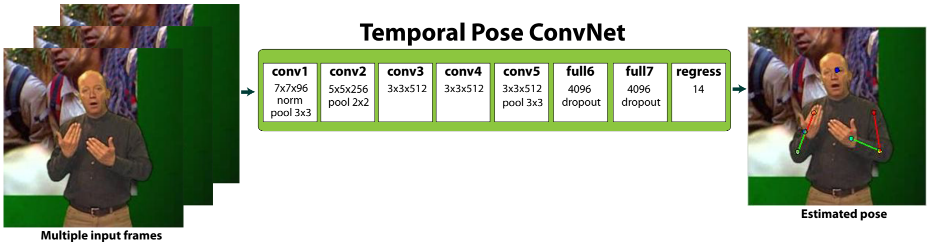

Fig. 1. Method overview. Given a set of input frames, the convolutional neural network regresses the positions of the head, shoulder, elbows and wrists.

Fig 1) we solve these issues by exploiting recent advances in deep convolutional neural networks (ConvNets) to accurately predict the pose without the need for foreground segmentation, and in real-time (100fps on a single GPU). Further, we show that our method implicitly learns constraints about the human kinematic chain, resulting in significantly better constrained pose estimates (*i.e.*, significantly smoother pose tracks with fewer serious prediction errors) than in previous work.

Many recent works have demonstrated the power of ConvNets in a wide variety of vision tasks – object classification and detection [7, 13, 17, 23], face recognition [20], text recognition [1, 8, 9], video action recognition [12, 18] and many more [6,14,16]. These networks comprise several layers of non-linear feature extractors and are therefore said to be 'deep' (in contrast to classic methods that are 'shallow'). Very recent works have also explored the use of ConvNets for estimating the human pose. Toshev and Szegedy [22] proposed to use a cascade of ConvNets to improve precision over a single network in unconstrained 2D pose estimation. Very recently, Tompson *et al.* [10, 21] proposed a hybrid architecture combining a ConvNet with a Markov Random Field-based spatial model. In this work we demonstrate that in gesture videos, a more computationally efficient conventional ConvNet alone outperforms previous work.

We evaluate our method on the BBC TV Signing dataset [3]. Our method achieves significantly better constrained pose estimates than the state of the art [3], without the need for hand-tuned foreground segmentation algorithms, and with over an order of magnitude faster computation speed.

2 Pose Estimation with ConvNets

In this paper we treat the task of estimating the pose as a regression problem. As the regressor we use a convolutional neural network, which consists of several stacked layers of convolutions and non-linearities. The input to the network is a set of RGB video frames, and the outputs of the last layer are the (x, y) coordinates of the upper-body joints.

We base our network architecture on that of Sermanet *et al.* [17] which has achieved excellent results on ImageNet Challenge 2013 object classification and

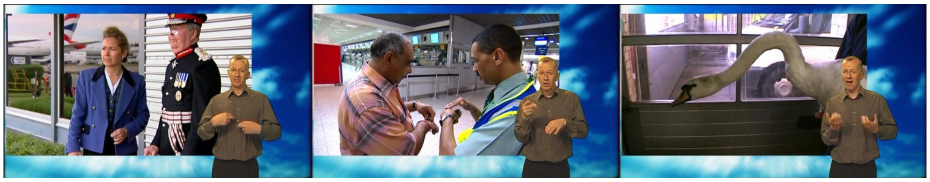

Fig. 2. Example frames from one video in the training set.

localisation tasks. The network is shown in Fig 1. Our network differs from Sermanet *et al*. in that we use multiple input frames and video-specific information to significantly improve generalisation performance, and we modify training time augmentation to better suit the task of pose estimation. We note that applying the ConvNets to a new problem domain (gesture videos) is all but straightforward, and requires taking into account many domain specifics.

We next give an overview of the architecture, followed by a discussion on the aspects in which our method differs from previous methods.

2.1 Architecture overview

Fig 1 shows the network architecture. It consists of five convolutional layers followed by three fully connected layers. A selection of convolutional layers are followed by pooling and local response normalisation layers, and the fully connected layers are regularised by dropout [13]. All hidden weight layers use a rectification activation function (RELU).

A generic ConvNet architecture is used due to its outstanding performance in image recognition tasks. In the experiments we show that using this generic architecture, along with a few important changes, we outperform previous work on a challenging video gesture pose dataset.

2.2 Pose regression

Regression layer. Our network is trained for regressing the location of the human upper-body joints. Instead of the softmax loss layer, found in the image classification ConvNets [13], we employ an l_2 loss layer, which penalises the l_2 distance between the pose predictions and ground truth. Since the absolute image coordinates of the people vary across videos, we first normalise the training set with regards to a bounding box. The bounding boxes are estimated using a face detector: the estimated face bounding boxes are scaled by a fixed scaler (learnt from the training data such that joints in all training frames are contained within the bounding boxes). In the image domain, we crop out the bounding box, and rescale it to a fixed height. In the human joint domain, we rescale accordingly, and in addition re-normalise the labels to the range $[0, 1]$. We found hyperparameter optimisation difficult without $[0, 1]$-normalised joints – in particular, the last fully-connected (regression) layer would require a different learning rate from other layers in order to converge.

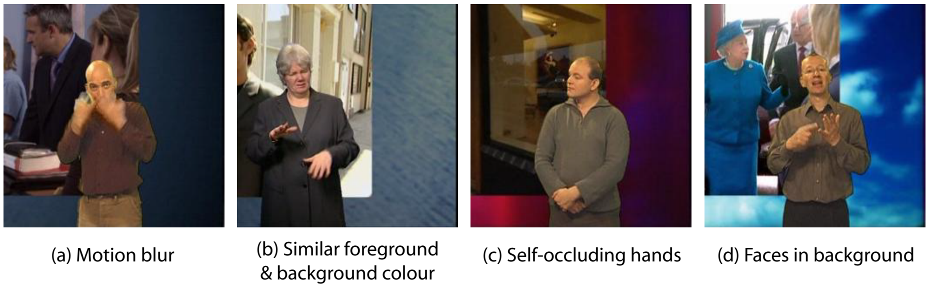

Fig. 3. Challenges in the BBC TV Signing dataset. (a) Motion blur removes much of the edges of the hand; (b) similar foreground & background colours render colour information less informative; (c) self-occluding hands makes the assignment of left/right hand ambiguous; (d) faces in the background renders face detection-based bounding box detection difficult.

We denote $(x, \mathbf{y})$ as a training example, where $\mathbf{y}$ stands for the coordinates of the k joints in the image x. Given normalised training data $N = (x, \mathbf{y})$ and a ConvNet regressor ϕ, the training objective becomes the task of estimating the the network weights λ:

$$\arg\min_{\lambda} \sum_{(x,\mathbf{y})\in N} \sum_{i=1}^{k} \|\mathbf{y}_i - \phi(x, \lambda)\|^2 \tag{1}$$

The ConvNet weights are optimised using backpropagation using the open-source Caffe framework [11].

Multiple input frames. To exploit the temporal information available in videos, our network is trained on multiple video frames. This is in contrast to CNN pose estimators in previous work, which typically operate on a single frame. This is done by inserting multiple frames (or their difference images) into the data layer colour channels. So for example, a network with three input frames contains 9 colour channels in its data layer. The network is then retrained from scratch. In practice, for the training of such a network to converge, input RGB values need to be rescaled by the number of input frames to preserve the dynamic range of the hyperparameters.

Video-specific learning. Pose estimation in long videos comes with its own challenges. Here we discuss how we overcome them.

The major difference to general RGB image pose estimation is that videos generally contain several frames of the same person, as depicted in Fig 2. Further, in the gesture communication scenario in particular, the person stands against a partially static background. We exploit this for an additional preprocessing step in our learning.

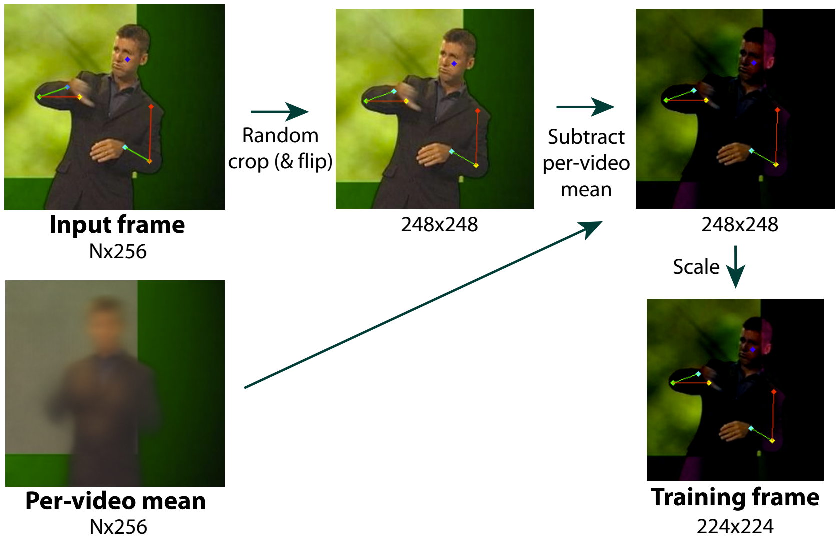

Fig. 4. Per-video mean and training augmentation. At training time, the training data is augmented with random crops and flips. A per-video mean is computed from a subset of frames to provide some invariance to different background colours. The per-video mean is obtained once per video, and can be computed on-the-fly in online pose estimation scenarios.

In particular, when training a generic network on videos without this preprocessing step, we noticed that the network would overfit to the static background behind the person. To alleviate this overfitting, we compute the mean image μ_V over 2,000 sampled frames for each video V in our training and testing datasets and subtract the video-specific mean from each input image: $x = x - \mu_V$ for frame x of video V. As shown in Fig 4, this removes the video-specific static background and yields an input representation for the ConvNet that generalises much better across different videos.

This method differs from previous static image ConvNets, which generally compute a mean image over the full dataset, and subtract the same mean from each input frame.

We note that this preprocessing can also be applied to test scenarios where there is no access to the full video. In those scenarios, the mean image can be computed over a small set of frames before the test frame. In practice we did not find this to cause a significant drop in prediction accuracy.

Training augmentation. Works on classification ConvNets commonly find that applying data augmentation (in the form of flips and crops) at training time increases performance [13]. In the classification tasks each image is typically associated with a single class; however, in our regression setting each image is associated with multiple target values dependent on the position of objects (body parts) in the image. We found that the level of data augmentation in the classification nets was too substantial for a regression task like ours.

Chatfield *et al.* [5] randomly crop (and randomly horizontally flip) a 224×224 subimage out of an image that has been resized so its smallest dimension is 256. This adds robustness to the absolute position of the object and improves generalisation in object recognition. However, when the input images are human bounding boxes, this crop is too substantial, and frequently crops off a part of the human body. One solution would be to scrap augmentation altogether. However, in experiments we found adding a small amount of invariance (as follows) to be helpful. We resize each input bounding box to height 256, randomly crop and flip a 248×248 image from it and update the joint positions accordingly. This results in each body part always being present in the image.

3 Implementation Details

Training. The training procedure is an adaptation of that of Krizhevsky *et al.* [13]. The network weights are learnt using mini-batch stochastic gradient descent with momentum set to 0.9. Each iteration samples 256 training frames randomly across the training videos and uses them as a mini-batch. The input frames are rescaled to height 256. A 248×248 sub-image (of the $N \times 256$ input image) is randomly cropped, randomly horizontally flipped and RGB jittered, and resized to 224×224. When re-training the ConvNet from scratch, the learning rate is set to 10^{-2}, and decreased to 10^{-3} at 80K iterations, to 10^{-4} after 90K iterations and stopped at 110K iterations. In the experiments in which we pretrain the weights on ImageNet ILSVRC-2012, learning rates are similarly decreased at 50K and 60K, and training is stopped at 70K iterations.

Testing. At test time, we crop the centre 248×248 of the input image, resize to 224×224 and feed forward through the network to obtain human joint location predictions. Test augmentation (*e.g.* computing the mean/median of predictions for 10 random image crops and flips, as done in classification ConvNet works) did not yield improved results over using the centre crop only.

Training time. Training was performed on a single NVIDIA GTX Titan GPU using a modified version of the Caffe framework [11]. Training the network from scratch took 3 days, while fine-tuning took 2 days.

Pretraining on ImageNet. In the evaluation section we also evaluate a setting where we pretrain the weights on ImageNet ILSVRC-2012 (rather than starting training from scratch), and fine-tune the weights on the new dataset. For those experiments, we use the publicly available “CNN-S” net provided by the authors of [5], which achieves 13.1% top-5 error on the ILSVRC-2012 test set.

4 Datasets

Experiments in this work are conducted on BBC sign language TV broadcasts. In addition to providing benchmarks on the original dataset, we also show experiments on an extended version of the dataset with an order of magnitude more training data.

Original BBC TV sign language broadcast dataset. This dataset consists of 20 TV broadcast videos overlaid with a person interpreting what is being spoken into British Sign Language (BSL) (see Fig 5). The videos, each between 0.5h–1.5h in length, contain content from a variety of TV programmes. All frames of the videos have been automatically assigned joint locations (which we use as ground truth for training) using a slow (and semi-automatic) but reliable tracker by Buehler *et al*. [2]. The full set of 20 videos are split into three disjoint sets: 13 videos for training, 2 for validation, 5 for testing. The test set videos contain different people wearing different clothing from those in the training and validation sets. 1,000 annotated frames (200 per video) of the test set have been manually annotated for evaluation purposes.

Extended BBC TV sign language broadcast dataset. This dataset contains 72 additional training videos, which are used to evaluate the benefits of additional training data. This dataset is combined with the original BBC TV dataset to yield a total of 92 videos (85 training, 2 validation and 5 testing). The frames of the new 72 videos are automatically assigned joint locations (used for ground truth in training) using the tracker of Charles *et al*. [3].

Foreground segmentations for both of these datasets are obtained automatically (but with some parameter tuning) using the co-segmentation algorithm of Charles *et al*. [3]. The output of the segmentation algorithm is an estimate of the person's silhouette (which can be noisy).

5 Evaluation

Experiments are conducted on the two BBC TV sign language datasets. We first present comparisons to alternative network architectures and preprocessing methods. We follow this by a comparison to the state of the art [3], both in terms of accuracy and computational performance.

5.1 Evaluation protocol and details

Evaluation protocol. In all pose estimation experiments we compare the estimated joints against frames with manual ground truth. We present results as graphs that plot accuracy vs distance from ground truth in pixels. A joint is deemed correctly located if it is within a set distance of d pixels from a marked joint centre in ground truth. Unless otherwise stated, the experiments use $d = 6$. Fig 5(top-left) shows the image scale.

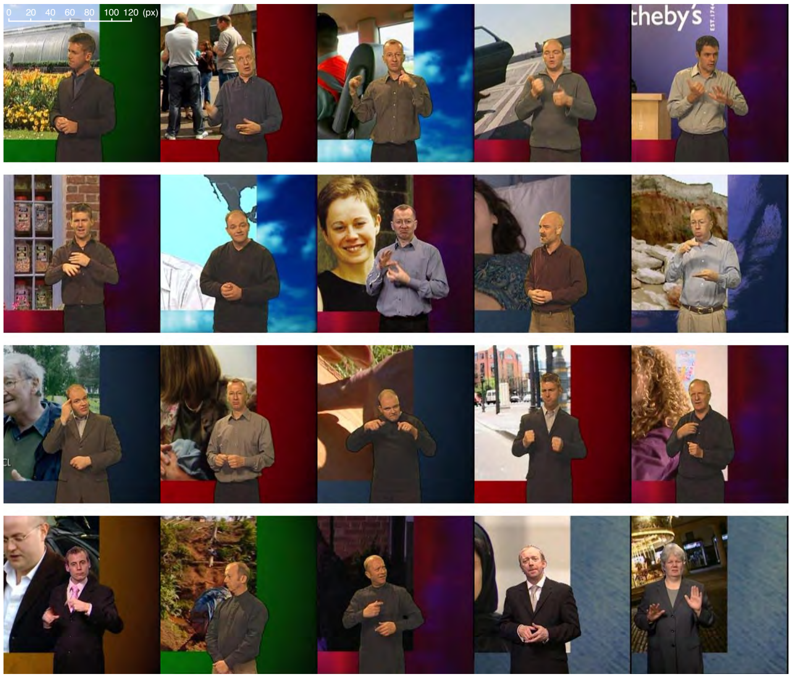

Fig. 5. Original BBC TV sign language dataset. The first three rows show the training and validation videos, and the last row shows the test videos. The upper-left figure gives a pixel scale.

Experimental details. All frames of the videos are used for training (with each frame randomly augmented as detailed above). The frames are randomly shuffled prior to training to present maximally varying input data to the network.

The hyperparameters (early stopping and weights for combining multiple nets, see below) are estimated using the validation set.

5.2 Evaluation of components

Table 1 shows comparisons to ConvNets with different number of input frames, input representations, levels of preprocessing and pretraining. We next discuss each of these results in detail.

Comparing ridge regression from ImageNet fully connected layer features. ConvNets trained on ImageNet have been shown to generalise extremely well to a wide variety of classification tasks [5, 16, 23]. Often the state of the art can be achieved by simply using the output from one of the fully connected layers as a feature. Here we investigate the performance of this approach for pose estimation.

In the first experiment, we extract features from the last fully connected layers of a network trained on ImageNet (a 4096 dimensional feature) applied to the original BBC TV broadcast dataset, and learn a ridge regressor. As shown in Table 1(row 1), this performs surprisingly poorly. This implies that these features that are extremely powerful for various real-world image classification tasks may not be quite as powerful when it comes to predicting precise locations of parts with high appearance variation in an image. One reason for this is likely that the network learns some location invariance.

Comparing to ImageNet pretraining/fine-tuning. In the second experiment, we first pretrain network weights on ImageNet ILSVRC-2012, and then fine-tune them on the BBC TV sign language dataset (Sect 3 provides details on pretraining). This performs better than ridge regression from the output of the fully connected layers, but still does not match the performance when the network is trained from scratch. This indicates that the two tasks (image classification and pose estimation) are sufficiently different to require considerably different weights.

Comparison to using a different mean image. In the third experiment, as a sanity check, we investigate using a single mean image for all training and testing. When using the standard ImageNet mean image, the average evaluation measure drops from 72.0% to 57.6%. The drop is caused by nearly completely failed tracking in some test videos. When investigating further, the reason turned out to be that the network overfitted to the backgrounds in the training data, and did not generalise to videos with different static backgrounds. This motivated the idea to use a per-video mean image, which removes the static background and so prevents the network from overfitting to it.

Comparison to a smaller ConvNet. In the fourth experiment, we turn to comparing our method to a smaller (and slightly faster) network (set up with same architecture as the "CNN-M" network in Chatfield *et al.* [5] – *i.e.*, same depth as our other network, 'CNN-S'", but with fewer parameters). This performs slightly worse than the larger network used in this work (60.4% vs 72.0%). This hints at that even deeper and larger networks could yield improved performance [19].

Comparison to no training augmentation. In the fifth experiment we test different levels of training augmentation. As shown in Table 1(row 3), training augmentation yields a small improvement over no training augmentation (using the centre crop of the image only), thanks to the added slight invariance in absolute position of the body inside the bounding box.

Comparisons with different training sets. In the sixth experiment we compare results when training either on the original, or extended, training dataset.

Training	Aug	Multi	Seg	Head	Wrists	Elbows	Shoulders	Average
Last only	✓			15.4	5.8	8.4	18.3	12.0
FT all	✓			95.6	44.0	53.6	80.8	68.5
Scratch				94.3	52.1	51.9	87.9	71.5
Scratch	✓			95.9	47.1	56.0	89.1	72.0
Scratch	✓	✓		95.6	50.1	58.1	89.5	73.3
Scratch	✓		✓	96.1	58.0	66.8	91.2	78.0
Scratch	✓	✓	✓	96.1	59.3	66.5	91.2	78.3

Table 1. Evaluation of different architectures. The evaluation measure is the percentage of predictions within 6 pixels from ground truth. 'Scratch/Last only/FT all' refer to training from scratch/training the last layer from scratch (keeping the rest of the ImNet-pretrained network fixed)/finetuning all layers of an ImNet-pretrained network; 'Aug' to training time augmentation; 'Multi' to using multiple input frames; and 'Seg' to using an input representation with the foreground pre-segmented.

The network trained on the extended dataset performs worse on shoulders (87.7% vs 89.1%), but better on wrists (53.0% vs 47.1%) and elbows (56.4% vs 56.0%), and slightly better on average (72.6% vs 72.0%). We hypothesise the better wrist performance is due to the network seeing a larger variety of poses, whereas the worse shoulder performance is due to less precise ground truth shoulder locations in the extended dataset.

Comparison to multi-frame net. In the seventh experiment we test the improvement from using multiple input frames. Table 1(row 5) shows a consistent performance improvement over wrists, elbows and shoulders, with a particularly noticeable improvement in wrist predictions. The head predictions are slightly worse, likely because the head is fairly stationary and hence does not benefit from the additional temporal information. Qualitatively, when visualising the predictions, the wrists are better localised and the output looks better smoothed.

The multi-frame network has two parameters: the number of input frames n (how many frames are used as input for each pose estimate) and the temporal spacing t between the input frames (time between each of the n input frames). In a parameter optimisation experiments we searched over $n = \{1, 3, 5\}$ and $t = \{1, 2, 3, 5, 8, 10, 15, 25\}$ on the validation set. $n = 3$ and $t = 1$ (three input frames with one-frame time spacing) were selected as the optimal parameters. We also explored using difference images (subtracting the current frame from the additional input frames), however this did not improve performance.

Weighted combination of nets. In the eighth experiment, we demonstrate a further improvement in performance by combining predictions from the two best-performing nets so far (with training augmentation that were trained from scratch – one single-frame and the other multi-frame). We define the prediction

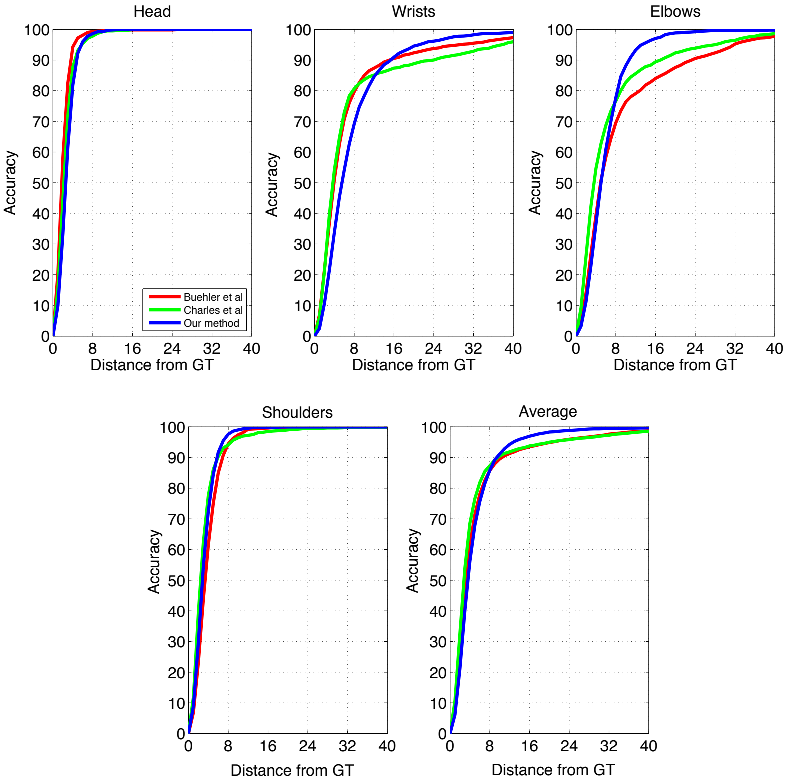

Fig. 6. Comparison to previous work. Comparison of pose estimation accuracy of the best-performing ConvNet versus the methods of Charles *et al*. [3] and Buehler *et al*. [2] (all trained and tested on the original dataset). Our method achieves much better constrained poses than previous work, without requiring any additional manual annotation (while [2] need per-video manual labelling, and [3] need manual tuning for the parameters of the foreground segmentation algorithm). Plots show accuracy per joint type (average over left and right body parts) as the allowed distance from manual ground truth is increased.

of each joint y_i as

$$y_i = \alpha_i C_1(x) + (1 - \alpha_i) C_2(x) \tag{2}$$

where $C_l(x)$ are the predictions of the two nets for input image x. The parameter α_i is learnt separately for each joint i on the validation set.

This combination yields the best performance of all the nets without foreground segmentations: 74.2% (vs 72.0% and 73.3% for the single-frame and multi-frame nets respectively).

Comparison to foreground-segmented input representation. In the ninth experiment we test our approach with the input representation of Charles *et al*., who pre-segment the foreground of the input frames using an algorithm with some manually tuned parameters, and black out the background. As shown in Table 1(row 6), even though our method does not require foreground segmentations, it does benefit from using them. On the downside, using them would require manual tuning of segmentation parameters and significantly slow down the runtime (from 100fps to around 5fps).

5.3 Comparison to previous work

Fig 6 presents a comparison of our method to previous work. Our method results on average in much better constrained poses and a significantly higher area under the curve, without requiring any of the video-specific manual segmentation algorithm tuning [3] or manual annotation [2] in previous work. We hypothesise that the better constrained poses are due to the ConvNet learning constraints for what poses a human body can perform (*i.e.*, the constraints of the human kinematic chain). While our method doesn't require any manual annotation (unlike previous work), our results in fact improve further if the input includes the foreground segmentations (as used in Charles *et al.*) – this is shown in Table 1.

Further, the ConvNet rarely predicts incorrect joint positions in the background (which is in contrast to previous work on this dataset, which reports 'catching the background' as the main challenge when not using a foreground segmentation algorithm [3] – Fig 8 shows examples of corrected frames). We conjecture that this is due to the high capacity of the model, which enables it to essentially learn a foreground segmentation and ignore any pixels in the background. In practice, this 'more constrained' estimation output manifests itself as significantly fewer serious prediction errors, and as much smoother pose tracks.

We achieve on average very similar pose estimates to previous work within the range of $0 - 8$ pixels from ground truth, and outperform from 8 pixels onwards. On elbows this marked improvement is clear from 8 pixels onwards, and for wrists from 14 pixels onwards. The performance on shoulders and head predictions is very similar to previous methods. The performance in the near range on wrists is slightly poorer, which we attribute to the lower resolution of our network (selected due to computational limitations) which makes accurate learning of the highly varying wrist positions more challenging. However, this is more than compensated for by (1) significantly better constrained predictions (see examples in Fig 8), (2) not needing a foreground segmentation algorithm that needs to be tuned manually, and (3) an order of magnitude faster prediction speed.

5.4 Computation time

Fig 7 shows a comparison of our method's computation time to previous work. The computation times are measured on a 2.4 GHz Intel i7 Quad Core CPU. We improve by an order of magnitude over previous methods using the same hardware. If we instead predict with a single GPU, performance increases by yet another order of magnitude. Note that the timing of our method assumes a cropped bounding box around the person. Our current method for computing

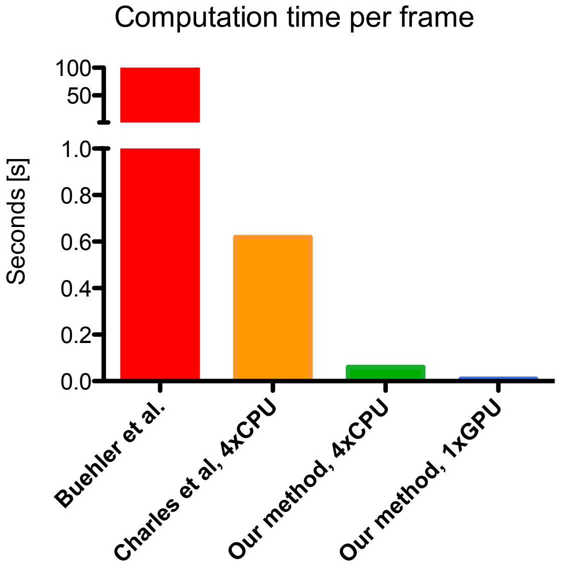

Fig. 7. Computation time. Comparison of computation times versus the methods of Charles *et al.* [3] and Buehler *et al.* [2]. Note that the reliable semi-automatic method of Buehler *et al.* is far off the scale as computing a pose for a single frame takes around 100s. Our method outperforms previous methods by over an order of magnitude using the same hardware. Using a single GPU instead of 4 CPUs increases speed by another order of magnitude.

this takes 0.06s on the CPU or 0.01 on the GPU using OpenCV's face detector. Even when taking these costs into account, our method is still over 5x faster on the CPU and 30x faster on the GPU.

6 Conclusion and Future Work

We have presented a ConvNet pose estimation method for gesture videos. The method produces much better constrained poses than previous work on the gesture dataset, is much faster and does not require a manually tuned foreground segmentation algorithm.

In the future we plan to investigate various avenues of improvement, including alternative ways of integrating temporal information from the videos into our networks, and training on significantly larger datasets.

Acknowledgements: We are grateful to Sophia Pfister for discussions. Financial support was provided by Osk. Huttunen Foundation and EPSRC grant EP/I012001/1.

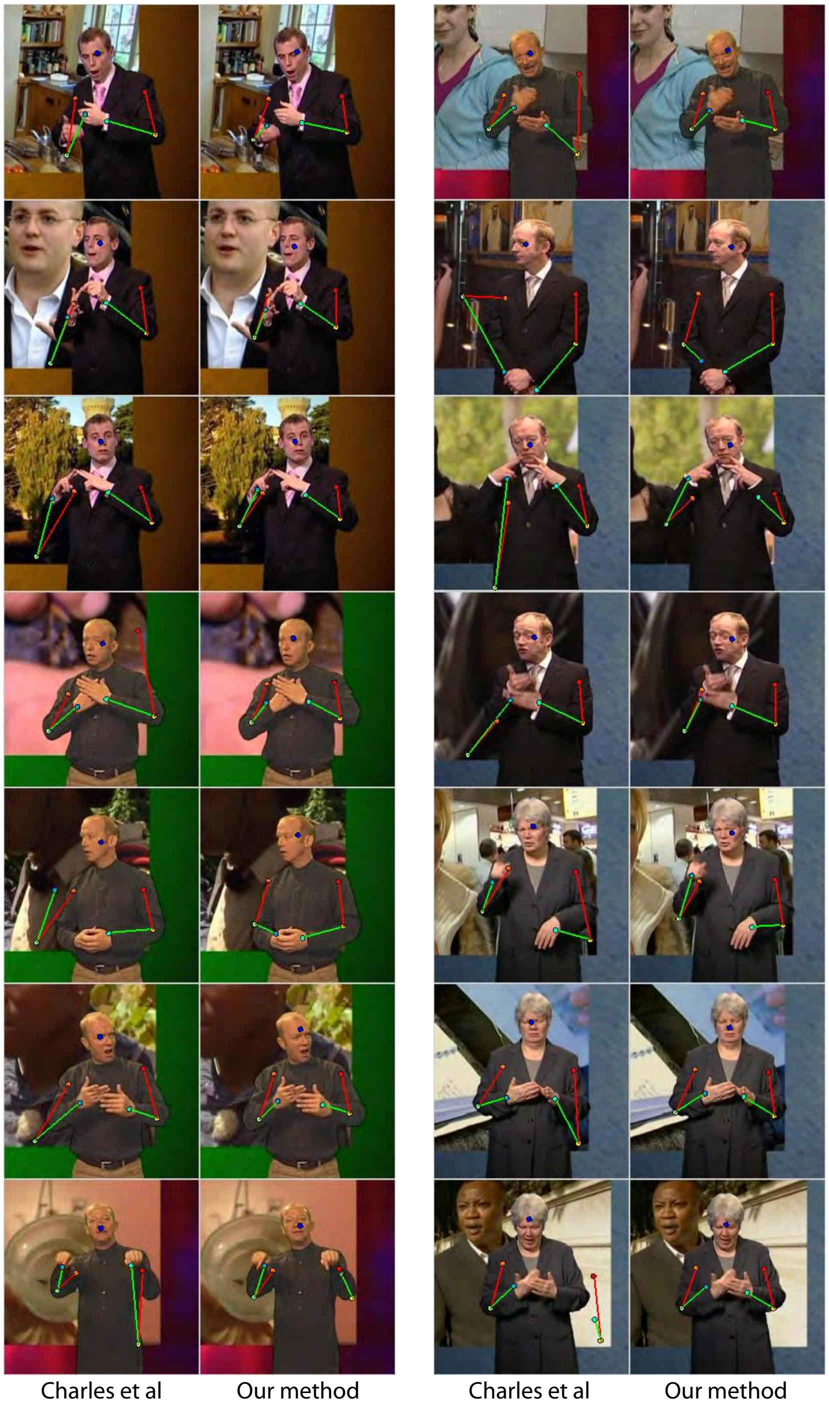

Fig. 8. Example test set frames comparing our estimates versus previous work [3]. The pose estimates of our method are much better localised.

References

1. Alsharif, O., Pineau, J.: End-to-end text recognition with hybrid hmm maxout models. In: ICLR (2014)
2. Buehler, P., Everingham, M., Huttenlocher, D.P., Zisserman, A.: Upper body detection and tracking in extended signing sequences. IJCV 95(2), 180–197 (2011)
3. Charles, J., Pfister, T., Everingham, M., Zisserman, A.: Automatic and efficient human pose estimation for sign language videos. IJCV (2013)
4. Charles, J., Pfister, T., Magee, D., Hogg, D., Zisserman, A.: Domain adaptation for upper body pose tracking in signed TV broadcasts. In: Proc. BMVC (2013)
5. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. BMVC (2014)
6. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. ICML (2014)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. CVPR (2014)
8. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. In: ICLR (2014)
9. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
10. Jain, A., Tompson, J., Andriluka, M., Taylor, G., Bregler, C.: Learning human pose estimation features with convolutional networks. ICLR (2014)
11. Jia, Y.: Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/` (2013)
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proc. CVPR (2014)
13. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
14. Osadchy, M., LeCun, Y., Miller, M.: Synergistic face detection and pose estimation with energy-based models. JMLR 8, 1197–1215 (2007)
15. Pfister, T., Charles, J., Everingham, M., Zisserman, A.: Automatic and efficient long term arm and hand tracking for continuous sign language TV broadcasts. In: Proc. BMVC (2012)
16. Razavian, S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. CVPR Workshops (2014)
17. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. ICLR (2014)
18. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. NIPS (2014)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proc. CVPR (2014)
21. Tompson, J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. NIPS (2014)

22. Toshev, A., Szegedy, C.: DeepPose: Human pose estimation via deep neural networks. CVPR (2014)
23. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. ECCV (2014)