arXiv:submit/1115263 [cs.CV] 16 Nov 2014

Efficient Object Localization Using Convolutional Networks

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, Christopher Bregler
New York University
tompson/goroshin/ajain/lecun/bregler@cims.nyu.edu

Figure 1: Our Model's Predicted Joint Positions on the MPII-human-pose database test-set[1]

Abstract

Recent state-of-the-art performance on human-body pose estimation has been achieved with Deep Convolutional Networks (ConvNets). Traditional ConvNet architectures include pooling layers which reduce computational requirements, introduce invariance and prevent over-training. These benefits of pooling come at the cost of reduced localization accuracy. We introduce a novel architecture which includes an efficient 'position refinement' model that is trained to estimate the joint offset location within a small region of the image. This refinement model is jointly trained in cascade with a state-of-the-art ConvNet model [21] to achieve improved accuracy in human joint location estimation. We show that the variance of our detector approaches the variance of human annotations on the FLIC [20] dataset and outperforms all existing approaches on the MPII-human-pose dataset [1].

1. Introduction

State-of-the-art performance on the task of human-body part localization has made significant progress in recent years. This has been in part due to the success of Deep-Learning architectures - specifically Convolutional Networks (ConvNets) [21, 14, 22, 5] - but also due to the availability of ever larger and more comprehensive datasets [1, 16, 20] (our model's predictions for difficult examples from [1] are shown in Figure 1).

A common characteristic of all ConvNet architectures used for human body pose detection to date is that they make use of internal strided-pooling layers. These layers reduce the spatial resolution by computing a summary statistic over a local spatial region (typically a max operation in the case of the commonly used Max-Pooling layer). The main motivation behind the use of these layers is to promote invariance to local input transformations (particularly translations) since their outputs are invariant to spatial location within the pooling region. This is particularly important for image classification where local image transformations obfuscate object identity. Therefore pooling plays a vital role in preventing over-training while reducing computational complexity for classification tasks.

The spatial invariance achieved by pooling layers comes at the price of limiting spatial localization accuracy. As such, by adjusting the amount of pooling in the network,

for localization tasks a trade-off is made between generalization performance, model size and spatial accuracy.

In this paper we present a ConvNet architecture for efficient localization of human skeletal joints in monocular RGB images that achieves high spatial accuracy without significant computational overhead. This model allows us to use increased amounts of pooling for computational efficiency, while retaining high spatial precision.

We begin by presenting a ConvNet architecture to perform coarse body part localization. This network outputs a low resolution, per-pixel heat-map, describing the likelihood of a joint occurring in each spatial location. We use this architecture as a platform to discuss and empirically evaluate the role of Max-pooling layers in convolutional architectures for dimensionality reduction and improving invariance to noise and local image transformations. We then present a novel network architecture that reuses hidden-layer convolution features from the coarse heat-map regression model in order to improve localization accuracy. By jointly-training these models, we show that our model outperforms recent state-of-the-art on standard human body pose datasets [1, 20].

2. Related Work

Following the seminal work of Felzenszwalb et al. [10] on 'Deformable Part Models' (DPM) for human-body-pose estimation, many algorithms have been proposed to improve on the DPM archiecture [2, 9, 23, 7]. Yang and Ramanan [23] propose a mixture of templates modeled using SVMs. Johnson and Everingham [15] propose more discriminative templates by using a cascade of body-part detectors.

Recently high-order DPM-based body-part dependency models have been proposed [18, 19, 11, 20]. Pishchulin [18, 19] use *Poselet* priors and a DPM model [3] to capture spatial relationships of body-parts. In a similar work, Gkioxari et al. [11] propose the *Armlets* approach which uses a semi-global classifier of part configurations. Their approach exhibits good performance on real-world data, however it is demonstrated only on arms. Sapp and Taskar [20] propose a multi-modal model including both holistic and local cues for coarse mode selection and pose estimation. A common characteristic to all these approaches is that they use hand-crafted features (edges, contours, HoG features and color histograms), which have been shown to have poor generalization performance and discriminative power in comparison to learned features (as in this work).

Today, the best performing algorithms for many vision tasks are based on convolutional networks (ConvNets). The current state-of-the-art methods for the task of human-pose estimation *in-the-wild* are also built using ConvNets [22, 13, 21, 14, 5]. The model of Toshev et al. [22] significantly output-performed state-of-art methods on the challenging 'FLIC' [20] dataset and was competitive on the 'LSP' [16] dataset. In contrast to our work, they formulate the problem as a direct (continuous) regression to joint location rather than a discrete heat-map output. However, their method performs poorly in the high-precision region and we believe that this is because the mapping from input RGB image to XY location adds unnecessary learning complexity which weakens generalization.

For example, direct regression does not deal gracefully with multi-modal outputs (where a valid joint is present in two spatial locations). Since the network is forced to produce a single output for a given regression input, the network does not have enough degrees of freedom in the output representation to afford small errors which we believe leads to over-training (since small outliers - due to for instance the presence of a valid body part - will contribute to a large error in XY).

Chen et al. [5] use a ConvNet to learn a low-dimensional representation of the input image and use an image dependent spatial model and show improvement over [22]. Tompson et al. [21] uses a multi-resolution ConvNet architecture to perform heat-map likelihood regression which they train jointly with a graphical model network to further promote joint consistency. In similar work, Jain et al. [14] also uses a multi-resolution ConvNet architecture, but they add motion features to the network input to further improve accuracy. Our Heat-Map regression model is largely inspired by both these works with improvements for better localization accuracy. The contributions of this work can be seen as an extension of the architecture of [21], where we attempt to overcome the limitations of pooling to improve the precision of the spatial locality.

In an unrelated application, Eigen et al. [8] predict depth by using a cascade of coarse to fine ConvNet models. In their work the coarse model is pre-trained and the model parameters are fixed when training the fine model. By contrast, in this work we suggest a novel shared-feature architecture which enables joint training of both models to improve generalization performance and which samples a subset of the feature inputs to improve runtime performance.

3. Coarse Heat-Map Regression Model

Inspired by the work of Tompson et al. [21], we use a multi-resolution ConvNet architecture (Figure 2) to implement a sliding window detector with overlapping contexts to produce a coarse heat-map output. Since our work is an extension of their model, we will only present a very brief overview of the architecture and explain our extensions to their model.

3.1. Model Architecture

The coarse heat-map regression model takes as input an RGB Gaussian pyramid of 3 levels (in Figure 2 only 2 lev-

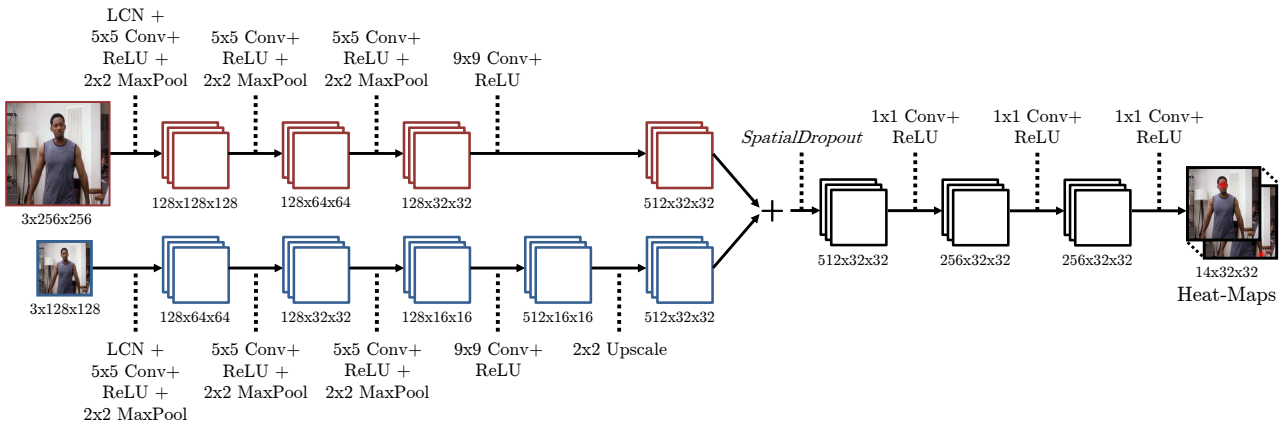


Figure 2: Multi-resolution Sliding Window Detector With Overlapping Contexts (model used on FLIC dataset)

els are shown for brevity) and outputs a heat-map for each joint describing the per-pixel likelihood for that joint occurring in each output spatial location. We use an input resolution of 320x240 and 256x256 pixels for the FLIC [20] and MPII [1] datasets respectively. The first layer of the network is a local-contrast-normalization (LCN) layer with the same filter kernel in each of the three resolution banks.

Each LCN image is then input to a 7 stage multi-resolution convolutional network (11 stages for the MPII dataset model). Due to the presence of pooling the heat-map output is at a lower resolution than the input image. It should be noted that the last 4 stages (or 3 stages for the MPII dataset model) effectively simulate a fully-connected network for a target input patch size (which is typically a much smaller context than the input image). We refer interested readers to [21] for more details.

3.2. *SpatialDropout*

We improve the model of [21] by adding an additional dropout layer before the first 1x1 convolution layer in Figure 2. The role of dropout is to improve generalization performance by preventing activations from becoming strongly correlated [12], which in turn leads to over-training. In the standard dropout implementation, network activations are "dropped-out" (by zeroing the activation for that neuron) during training with independent probability p_{drop}. At test time all activations are used, but a gain of $1 - p_{\text{drop}}$ is multiplied to the neuron activations to account for the increase in expected bias.

In initial experiments, we found that applying standard dropout (where each convolution feature map activation is "dropped-out" independently) before the 1×1 convolution layer generally increased training time but did not prevent over-training. Since our network is fully convolutional and natural images exhibit strong spatial correlation, the feature map activations are also strongly correlated, and in this setting standard dropout fails.

Standard dropout at the output of a 1D convolution is illustrated in Figure 3. The top two rows of pixels represent the convolution kernels for feature maps 1 and 2, and the bottom row represents the output features of the previous layer. During back-propagation, the center pixel of the W_2 kernel receives gradient contributions from both f_{2a} and f_{2b} as the convolution kernel W_2 is translated over the input feature F_2. In this example f_{2b} was randomly dropped out (so the activation was set to zero) while f_{2a} was not. Since F_2 and F_1 are the output of a convolution layer we expect f_{2a} and f_{2b} to be strongly correlated: i.e. $f_{2a} \approx f_{2b}$ and $de/df_{2a} \approx de/df_{2b}$ (where e is the error function to minimize). While the gradient contribution from f_{2b} is zero, the strongly correlated f_{2a} gradient remains. In essence, the effective learning rate is scaled by the dropout probability p, but independence is not enhanced.

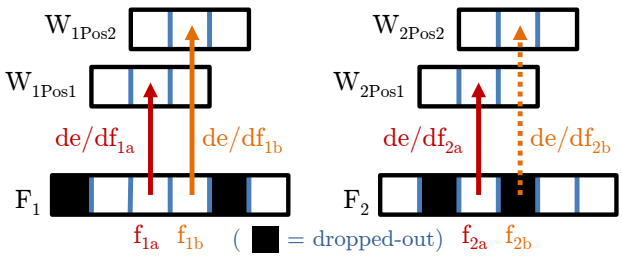


Figure 3: Standard Dropout after a 1D convolution layer

Instead we formulate a new dropout method which we call *SpatialDropout*. For a given convolution feature tensor of size $n_{\text{feats}} \times \text{height} \times \text{width}$, we perform only n_{feats} dropout trials and extend the dropout value across the entire feature map. Therefore, adjacent pixels in the dropped-out feature map are either all 0 (dropped-out) or all active as illustrated

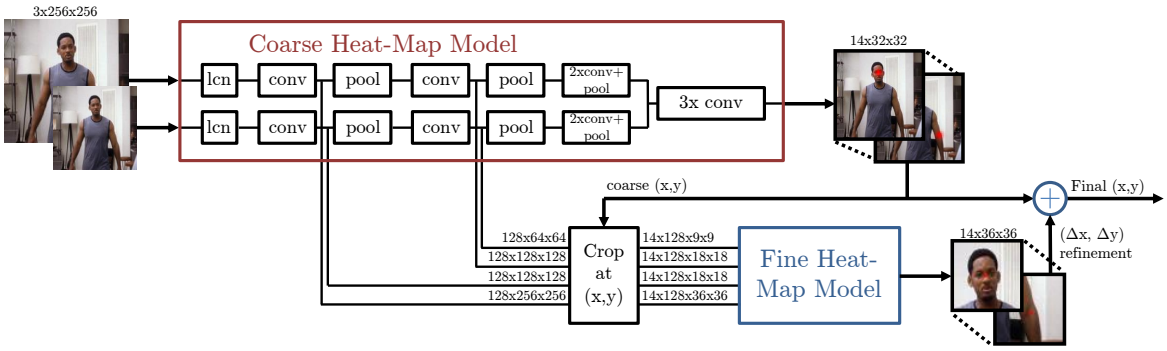


Figure 4: Overview of our Cascaded Architecture

in Figure 5. We have found this modified dropout implementation improves performance, especially on the FLIC dataset, where the training set size is small.

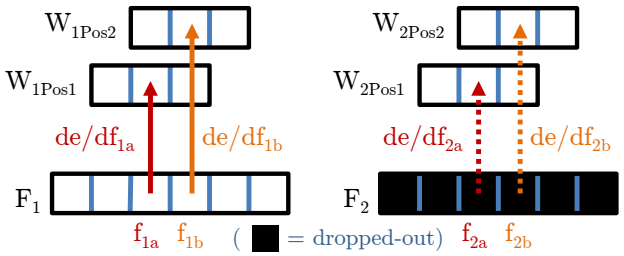


Figure 5: *SpatialDropout* after a 1D convolution layer

3.3. Training and Data Augmentation

We train the model in Figure 2 by minimizing the Mean-Squared-Error (MSE) distance of our predicted heat-map to a target heat-map. The target is a 2D Gaussian of constant variance ($\sigma \approx 1.5$ pixels) centered at the ground-truth (x, y) joint location. The objective function is:

$$E_1 = \frac{1}{N} \sum_{j=1}^{N} \sum_{xy} \left\| H'_j(x,y) - H_j(x,y) \right\|^2 \qquad (1)$$

Where H'_j and H_j are the predicted and ground truth heat-maps respectively for the jth joint.

During training, each input image is randomly rotated ($r \in [-20°, +20°]$), scaled ($s \in [0.5, 1.5]$) and flipped (with probability 0.5) in order to improve generalization performance on the validation-set.

Many images contain multiple people while only a single person is annotated. To enable inference of the target person's annotations at test time, both the FLIC and MPII datasets include an approximate torso position. Since our sliding-window detector will detect all joint instances in a single frame indiscriminately, we incorporate this torso information by implementing the MRF-based spatial model of Tompson et al. [21].

4. Fine Heat-Map Regression Model

In essence, the goal of this work is to recover the spatial accuracy lost due to pooling of the model in Section 3.1 by using an additional ConvNet to refine the localization result of the coarse heat-map. However, unlike a standard cascade of models, as in the work of Toshev et al. [22], we reuse existing convolution features. This not only reduces the number of trainable parameters in the cascade, but also acts as a regularizer for the coarse heat-map model since the coarse and fine models are trained jointly.

4.1. Model Architecture

The full system architecture is shown in Figure 4. It consists of the heat-map-based parts model from Section 3.1 for coarse localization, a module to sample and crop the convolution features at a specified (x, y) location for each joint, as well as an additional convolutional model for fine tuning.

Joint inference from an input image is as follows: we forward-propagate (FPROP) through the coarse heat-map model then infer all joint (x, y) locations from the maximal value in each joint's heat-map. We then use this coarse (x, y) location to sample and crop the first 2 convolution layers (for all resolution banks) at each of the joint locations. We then FPROP these features through a fine heat-map model to produce a $(\Delta x, \Delta y)$ offset within the cropped sub-window. Finally, we add the position refinement to the coarse location to produce a final (x, y) localization for each joint.

Figure 6 shows the crop module functionality for a single joint. We simply crop out a window centered at the coarse joint (x, y) location in each resolution feature map, however we do so by keeping the contextual size of the window constant by scaling the cropped area at each higher resolution

level. Note that back-propagation (BPROP) through this module from output feature to input feature is trivial; output gradients from the cropped image are simply added to the output gradients of the convolution stages in the coarse heat-map model at the sampled pixel locations.

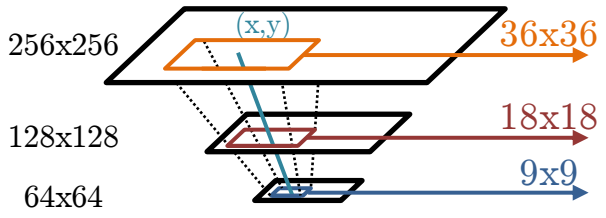


Figure 6: Crop module functionality for a single joint

The fine heat-map model is a Siamese network [4] of 7 instances (14 for the MPII dataset), where the weights and biases of each module are shared (i.e. replicated across all instances and updated together during BPROP). Since the sample location for each joint is different, the convolution features do not share the same spatial context and so the convolutional sub-networks must be applied to each joint independently. However, we use parameter sharing amongst each of the 7 instances to substantially reduce the number of shared parameters and to prevent over-training. At the output of each of the 7 sub-networks we then perform a 1x1 Convolution, with no weight sharing to output a detailed-resolution heat-map for each joint. The purpose of this last layer is to perform the final detection for each joint.

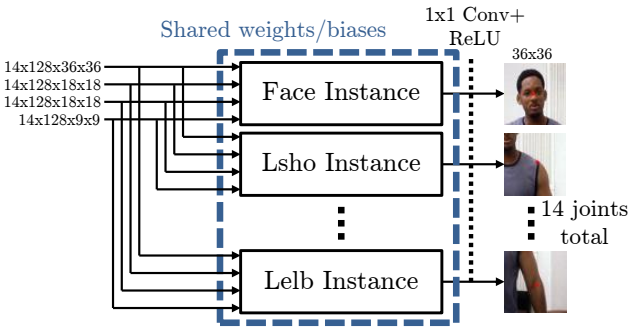


Figure 7: Fine heat-map model: 14 joint Siamese network

Note we are potentially performing redundant computations in the Siamese network. If two cropped sub-windows overlap and since the convolutional weights are shared, the same convolution maybe applied multiple times to the same spatial locations. However, we have found in practice this is rare. Joints are infrequently co-located, and the spatial context size is chosen such that there is little overlap between cropped sub-regions (note that the context of the cropped images shown in Figures 4 and 8 are exaggerated for clarity).

Each instance of the sub-network in Figure 7 is a ConvNet of 4 layers, as shown in Figure 8. Since the input images are different resolutions and originate from varying depths in the coarse heat-map model, we treat the input features as separate resolution banks and apply a similar architecture strategy as used in Section 3.1. That is we apply the same size convolutions to each bank, upscale the lower-resolution features to bring them into canonical resolution, add the activations across feature maps then apply 1x1 convolutions to the output features.

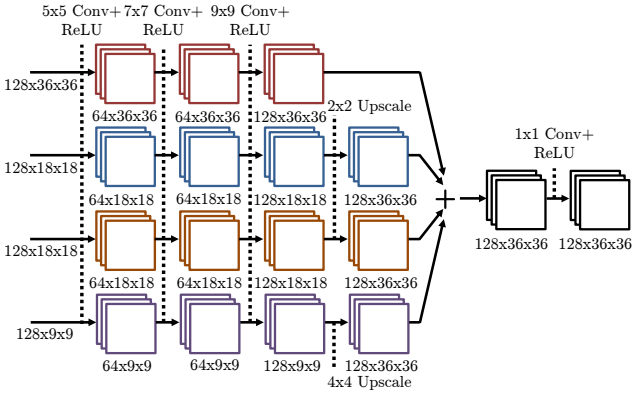


Figure 8: The fine heat-map network for a single joint

It should be noted that this cascaded architecture can be extended further as is possible to have multiple cascade levels each with less and less pooling. However, in practice we have found that a single layer provides sufficient accuracy, and in particular within the level of label noise on the FLIC dataset (as we show in Section 5).

4.2. Joint Training

We jointly train both the coarse and fine heat-map models together by minimizing the following objective function:

$$E_2 = E_1 + \lambda \frac{1}{N} \sum_{j=1}^{N} \sum_{x,y} \left\| G'_j(x,y) - G_j(x,y) \right\|^2 \quad (2)$$

Where G' and G are the set of predicted and ground truth heat-maps respectively for the fine heat-map model, λ is a constant used to trade-off the relative importance of both sub-tasks and E_1 is the coarse heat-map model contribution from Equation 1. We treat λ as another network hyper-parameter and is chosen to optimize performance over our validation set (we use $\lambda = 0.1$). Ideally, a more direct optimization function would attempt to measure the $argmax$

of both heat-maps and therefore directly minimize the final (x, y) prediction. However, since the $argmax$ function is not differentiable we instead reformulate the problem as a regression to a set of target heat-maps and minimize the distance to those heat-maps.

5. Results

Our ConvNet architecture was implemented within the Torch7 [6] framework and evaluation is performed on the FLIC [20] and MPII-Human-Pose [1] datasets. The FLIC dataset consists of 3,987 training examples and 1,016 test examples of still scenes from Hollywood movies annotated with upper-body joint labels. Since the poses are predominantly front-facing and upright, FLIC is considered to be less challenging than more recent datasets. However the small number of training examples makes the dataset a good indicator for generalization performance. On the other-hand the MPII dataset is very challenging and it includes a wide variety of full-body pose annotations within the 28,821 training and 11,701 test examples. For evaluation of our model on the FLIC dataset we use the standard PCK measure proposed by [20] and we use the PCKh measure of [1] for evaluation on the MPII dataset.

Figure 9 shows the PCK test-set performance of our coarse heat-map model (Section 3.1) when various amounts of pooling are used within the network (keeping the number of convolution features constant). Figure 9 results show quite clearly the expected effect of coarse quantization in (x, y) and therefore the impact of pooling on spatial precision; when more pooling is used the performance of detections within small distance thresholds is reduced.

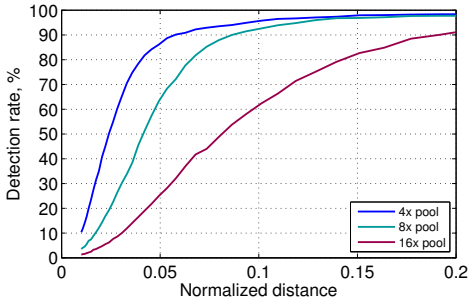


Figure 9: Pooling impact on FLIC test-set Average Joint Accuracy for the coarse heat-map model

For joints where the ground-truth label is ambiguous and difficult for the human mechanical-turkers to label, we do not expect our cascaded network to do better than the expected variance in the user-generated labels. To measure this variance (and thus estimate the upper bound of performance) we performed the following informal experiment: we showed 13 users 10 random images from the FLIC training set with annotated ground-truth labels as a reference so that the users could familiarize themselves with the desired anatomical location of each joint. The users then annotated a consistent set of 10 random images from the FLIC test-set for the face, left-wrist, left-shoulder and left-elbow joints. Figure 10 shows the resultant joint annotations for 2 of the images.

	Face	Shoulder	Elbow	Wrist
Label Noise (10 images)	0.65	2.46	2.14	1.57
This work 4x (test-set)	1.09	2.43	2.59	2.82
This work 8x (test-set)	1.46	2.72	2.49	3.41
This work 16x (test-set)	1.45	2.78	3.78	4.16

Table 1: σ of (x, y) pixel annotations on FLIC test-set images (at 360×240 resolution)

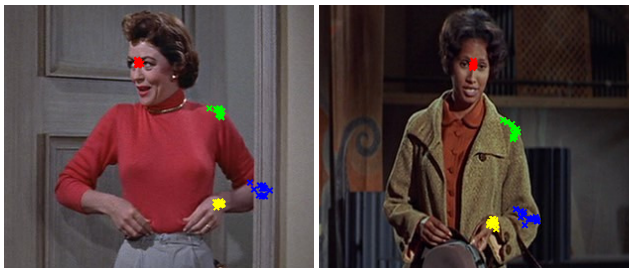


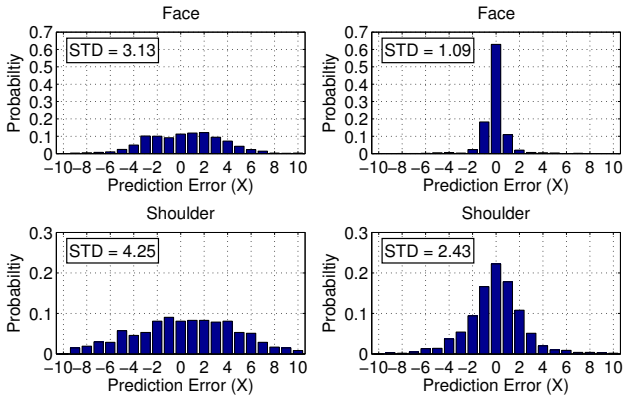
Figure 10: User generated joint annotations

To estimate joint annotation noise we calculate the standard deviation (σ) across user annotations in x for each of the 10 images separately and then average the σ across the 10 sample images to obtain an aggregate σ for each joint. Since we down-sample the FLIC images by a factor of 2 for use with our model we divide the σ values by the same down-sample ratio. The result is shown in Table 1.

The histogram of the coarse heat-map model pixel error (in the x dimension) on the FLIC test-set when using an 8x internal pooling is shown in Figure 11a (for the face and shoulder joints). For demonstration purposes, we quote the error in the pixel coordinates of the input image to the network (which for FLIC is 360×240), not the original resolution. As expected, in these coordinates there is an approximately uniform uncertainty due to quantization of the heat-map within -4 to +4 pixels. In contrast to this, the histogram of the cascaded network is shown in Figure 11b and is close to the measured label noise[1].

PCK performance on FLIC for face and wrist are shown in Figures 12a and 12b respectively. For the face, the per-

[1] When calculating σ for our model, we remove all outliers with error $>$ 20 and error < -20. These outliers represent samples where our weak spatial model chose the wrong person's joint and so do not represent an accurate indication of the spatial accuracy of our model.



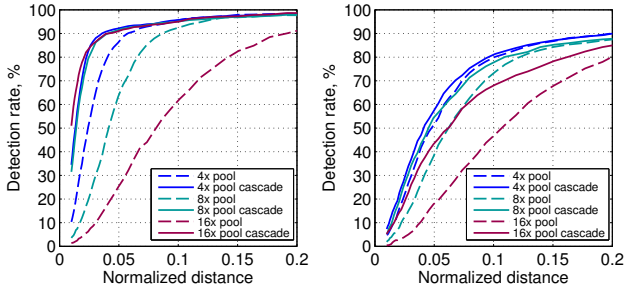
(a) Coarse model only (b) Cascaded model

Figure 11: Histogram of X error on FLIC test-set

	4x pool	8x pool	16x pool
Coarse-Model	140.0	74.9	54.7
Fine-Model	17.2	19.3	15.9
Cascade	157.2	94.2	70.6

Table 2: Forward-Propagation time (seconds) for each of our FLIC trained models

formance improvement is significant, especially for the $8\times$ and $16\times$ pooling part models. The FPROP time for a single image (using an Nvidia-K40 GPU) for each of our models is shown in Table 2; using the $8\times$ pooling cascaded network, we are able to perform close to the level of label noise with a significant improvement in computation time over the $4\times$ network.

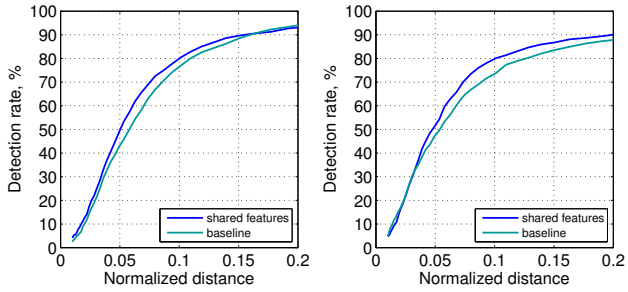


(a) Face (b) Wrist

Figure 12: Performance improvement from cascaded model

The performance improvement for wrist is also significant but only for the $8\times$ and $16\times$ pooling models. Our empirical experiments suggest that wrist detection (as one of the hardest to detect joints) requires learning features with a large amount of spatial context. This is because the wrist joint undergoes larger amounts of skeletal deformation than the shoulder or face, and typically has high input variability due to clothing and wrist accessories. Therefore, with limited convolution sizes and sampling context in the fine heat-map regression network, the cascaded network does not improve wrist accuracy beyond the coarse approximation.

To evaluate the effectiveness of the use of shared features for our cascaded network we trained a fine heat-map model that takes it's inputs from a cropped version of the input image rather than the first and second layer convolution feature maps of our coarse heat-map model. This comparison model is a greedily-trained cascade, where the coarse and fine models are trained independently. Additionally, since the network in Figure 4 has a higher capacity than the comparison model, we add an additional convolution layer such that the number of trainable parameters is the same. Figures 13a and 13b shows that our 4x pooling network outperforms this comparison model. We attribute this to the regularizing effect of joint training; the fine heat-map model term in the objective function prevents over-training of the coarse model and vice-versa.



(a) Elbow (b) Wrist

Figure 13: FLIC performance of our shared-features cascade vs an independently trained cascade

Figure 14 compares our detector's PCK performance averaged for the wrist and elbow joints with previous work. Our model outperforms the previous state-of-the-art results by Tompson et al. [21] for large distances, due to our use of *SpatialDropout*. In the high precision region the cascaded network is able to out-perform all state-of-the-art by a significant margin. The PCK performance at a normalized distance of 0.05 for each joint is shown in Table 3.

Finally, Figure 15 shows the PCKh performance of our model on the MPII human pose dataset. Similarity, table 4 shows a comparison of the PCKh performance of our model and previous state-of-the-art at a normalized distance of 0.5. Our model out-performs all existing methods by a considerable margin.

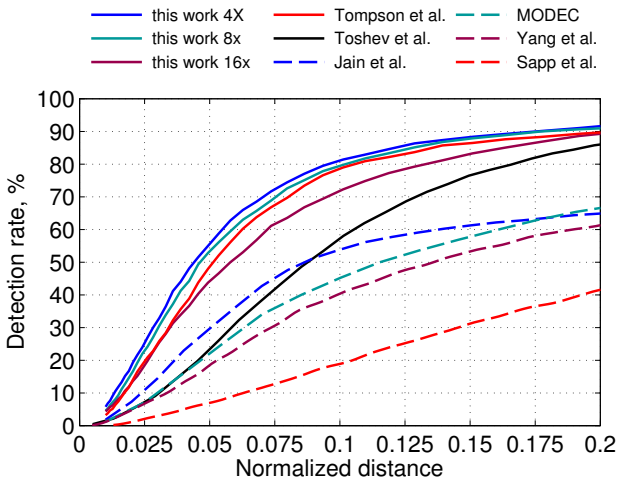


Figure 14: FLIC - average PCK for wrist and elbow

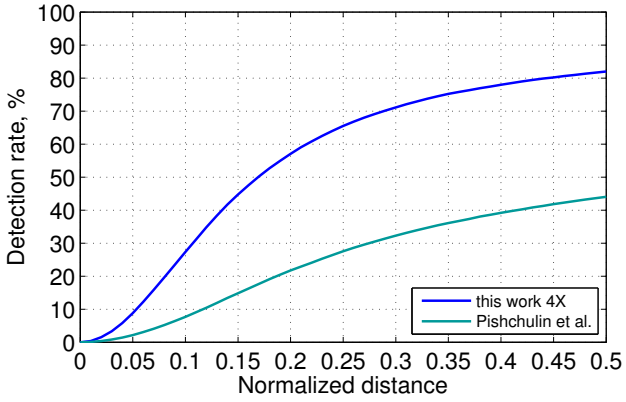


Figure 15: MPII - average PCKh for all joints

6. Conclusion

Though originally developed for the task of classification [17], Deep Convolutional Networks have been successfully applied to a multitude of other problems. In classification all variability except the object identity is suppressed. On the other hand, localization tasks such as human body pose estimation often demand a high degree of spatial precision. In this work we have shown that the precision lost due to pooling in traditional ConvNet architectures can be recovered efficiently while maintaining the computational benefits of pooling. We presented a novel cascaded architecture that combined fine and coarse scale convolutional networks, which achieved new state-of-the-art results on the FLIC [20] and MPII-human-pose[1] datasets.

7. Acknowledgements

This research was funded in part by Google and by the Office of Naval Research ONR Award N000141210327. We would also like the thank all the contributors to Torch7 [6], particularly Soumith Chintala, for all their hard work.

	Head	Shoulder	Elbow	Wrist
Yang et al.	-	-	22.6	15.3
Sapp et al.	-	-	6.4	7.9
Eichner et al.	-	-	11.1	5.2
MODEC et al.	-	-	28.0	22.3
Toshev et al.	-	-	25.2	26.4
Jain et al.	-	42.6	24.1	22.3
Tompson et al.	90.7	70.4	50.2	55.4
This work 4x	**92.6**	73.0	**57.1**	**60.4**
This work 8x	92.1	**75.8**	55.6	56.6
This work 16x	91.6	73.0	47.7	45.5

Table 3: Comparison with prior-art on FLIC (PCK @ 0.05)

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Upper Body	Full Body
Gkioxari et al.	-	36.3	26.1	15.3	-	-	-	25.9	-
Sapp & Taskar	-	38.0	26.3	19.3	-	-	-	27.9	-
Yang & Ramanan	73.2	56.2	41.3	32.1	36.2	33.2	34.5	43.2	44.5
Pishchulin et al.	74.2	49.0	40.8	34.1	36.5	34.4	35.1	41.3	44.0
This work 4x	**96.0**	**91.9**	**83.9**	**77.7**	**80.9**	**72.2**	**64.8**	**84.5**	**82.0**

Table 4: Comparison with prior-art: MPII (PCKh @ 0.5)

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. 2014. 1, 2, 3, 6, 8

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009. 2

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[4] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 1993. 5

[5] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *NIPS*, 2014. 1, 2

[6] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 6, 8

[7] M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR'13*. 2

[8] C. P. David Eigen and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, 2014. 2

[9] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009. 2

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[11] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR'13*. 2

[12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 3

[13] A. Jain, J. Tompson, M. Andriluka, G. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. In *ICLR*, 2014. 2

[14] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. *ACCV*, 2014. 1, 2

[15] S. Johnson and M. Everingham. Learning Effective Human Pose Estimation from Inaccurate Annotation. In *CVPR'11*. 2

[16] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 1, 2

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998. 8

[18] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR'13*. 2

[19] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV'13*. 2

[20] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 1, 2, 3, 6, 8

[21] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Join training of a convolutional network and a graphical model for human pose estimation. *NIPS*, 2014. 1, 2, 3, 4, 7

[22] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1, 2, 4

[23] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR'11*. 2