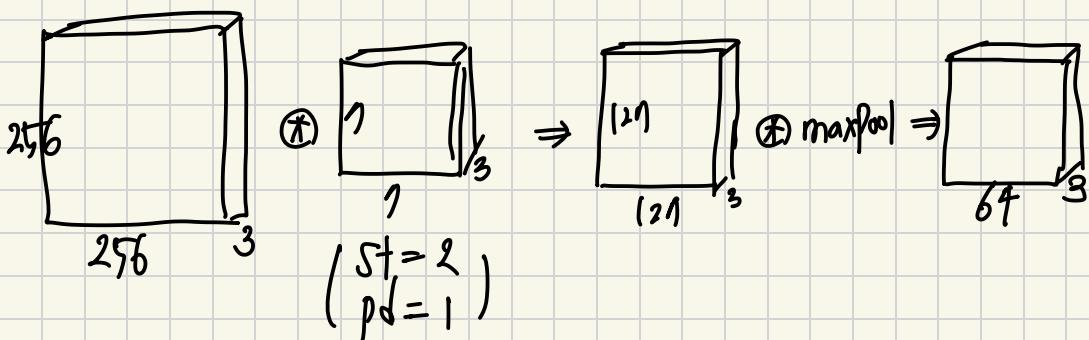
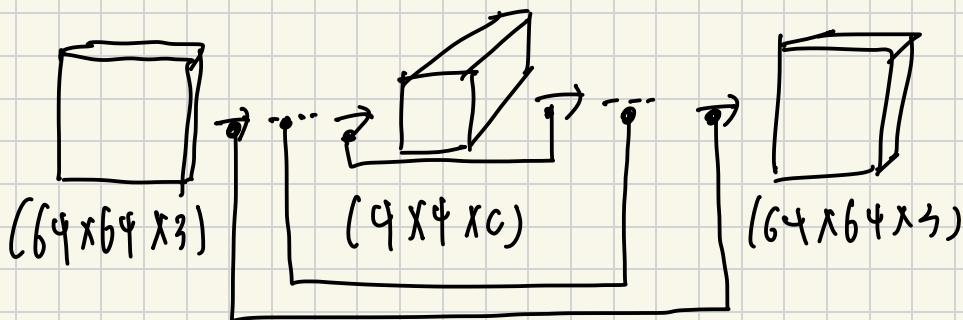


* 간단한 module 정리

$$\frac{256 - 1 + 2}{2} + 1 = \frac{251}{2} + 1 = 125.5 \Rightarrow 125$$

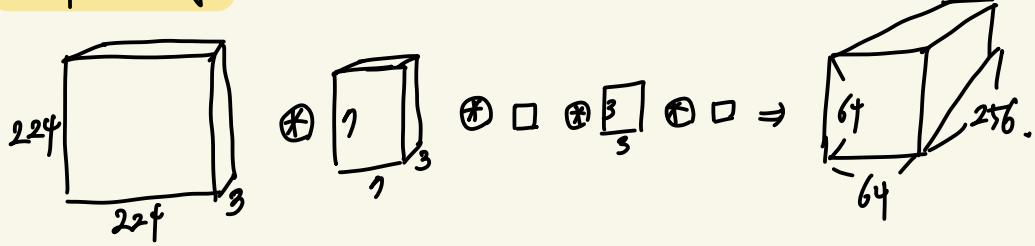


\therefore GPN 할당 iSSVE3 Image 를 $(256 \times 256) \Rightarrow (64 \times 64)$

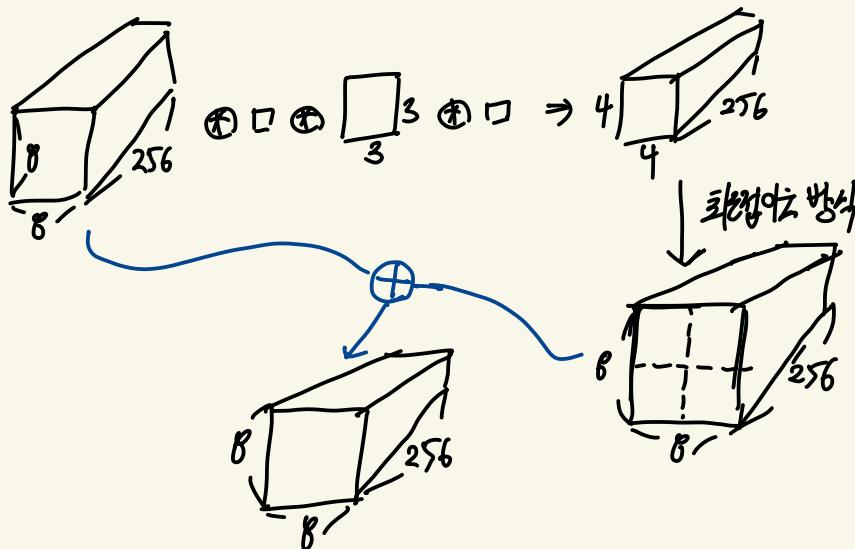
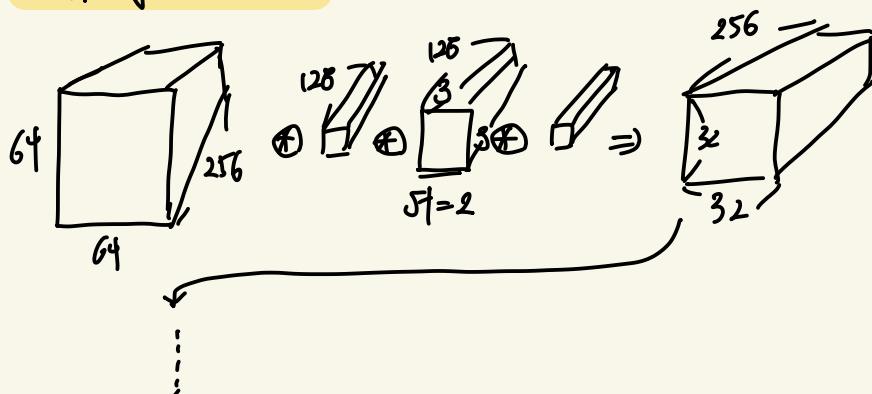


$$\underbrace{(64 \times 64 \times 3)}_{\rightarrow} \rightarrow (64 \times 64 \times 128) \rightarrow (32 \times 32 \times 128) \rightarrow \underbrace{(32 \times 32 \times 256)}$$

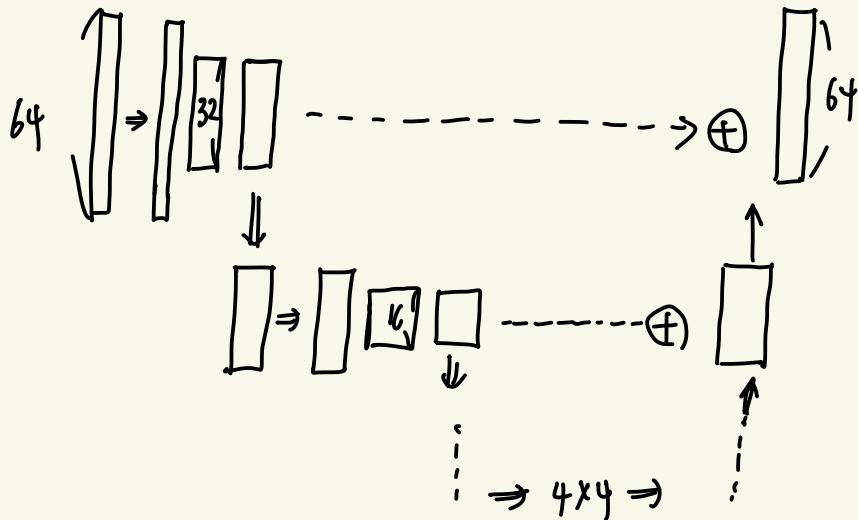
① Input Image



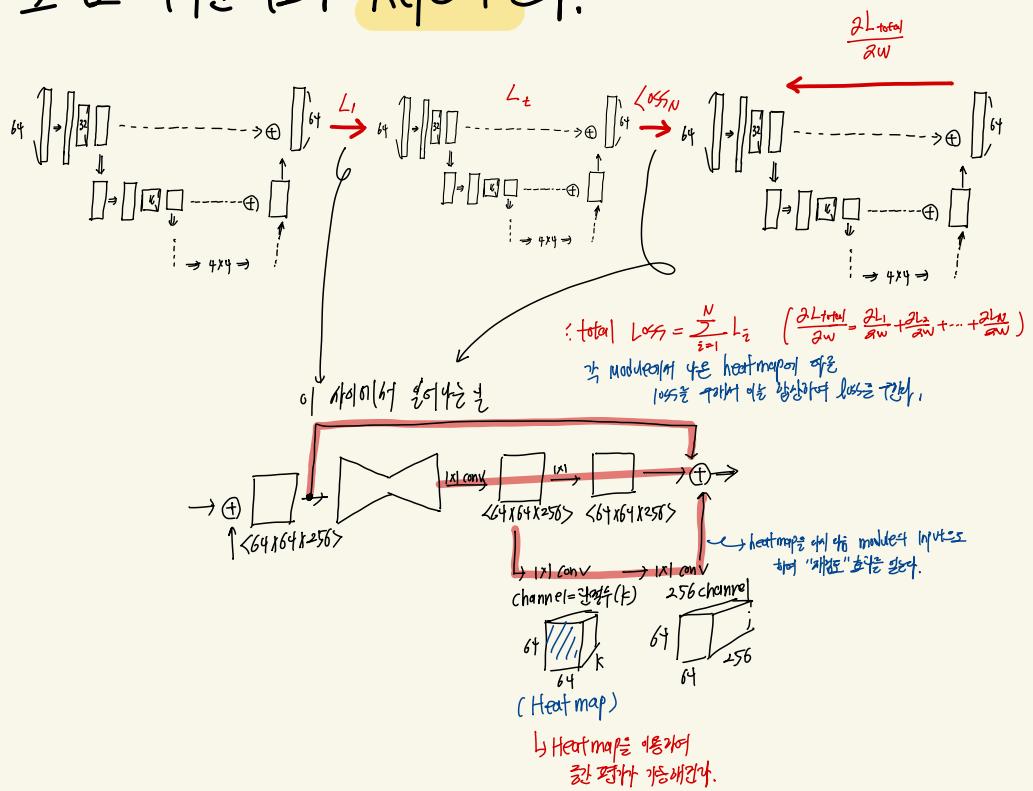
② Hourglass Module



∴ 결과 VNet 같은 구조를 갖는다.



그리고 어떤 구조가 Repeat되.



Stacked Hourglass Networks for Human Pose Estimation

Alejandro Newell, Kaiyu Yang, and Jia Deng

University of Michigan, Ann Arbor
 {alnewell, yangky, jiadeng}@umich.edu

Abstract. This work introduces a novel convolutional network architecture for the task of human pose estimation. Features are processed across all scales and consolidated to best capture the various spatial relationships associated with the body. We show how repeated bottom-up, top-down processing used in conjunction with intermediate supervision is critical to improving the performance of the network. We refer to the architecture as a “stacked hourglass” network based on the successive steps of pooling and upsampling that are done to produce a final set of predictions. State-of-the-art results are achieved on the FLIC and MPII benchmarks outcompeting all recent methods.

Keywords: Human Pose Estimation : (top-down & bottom-up) 45개의 18개의
 human pose estimation CNN Model! Stacked Hourglass Model!

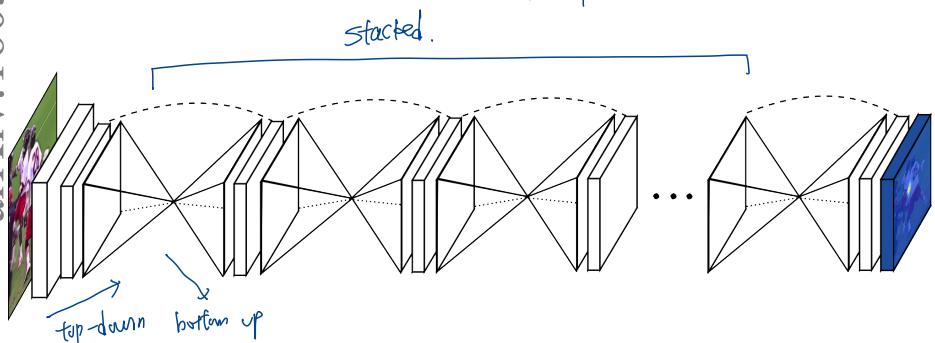


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

1 Introduction

A key step toward understanding people in images and video is accurate pose estimation. Given a single RGB image, we wish to determine the precise pixel location of important keypoints of the body. Achieving an understanding of a person’s posture and limb articulation is useful for higher level tasks like action recognition, and also serves as a fundamental tool in fields such as human-computer interaction and animation.

As a well established problem in vision, pose estimation has plagued researchers with a variety of formidable challenges over the years. A good pose estimation system must be robust to occlusion and severe deformation, successful on rare and novel poses, and invariant to changes in appearance due to factors like clothing and lighting. Early work tackles such difficulties using **robust image features** and **sophisticated structured prediction** [1–9]: the former is used to produce local interpretations, whereas the latter is used to infer a globally consistent pose. ↳ ① 각 부위별로 특징을 인식 → ② 각 부위별로 특징을 학습하여 합침.

This conventional pipeline, however, has been greatly reshaped by convolutional neural networks (**ConvNets**) [10–14], a main driver behind an explosive rise in performance across many computer vision tasks. Recent pose estimation systems [15–20] have universally adopted ConvNets as their main building block, largely replacing hand-crafted features and graphical models; this strategy has yielded drastic improvements on standard benchmarks [1, 21, 22].

We continue along this trajectory and introduce a novel “stacked hourglass” network design for predicting human pose. The network captures and consolidates information **across all scales of the image**. We refer to the design as an hourglass based on our visualization of the steps of pooling and subsequent up-sampling used to get the final output of the network. Like many convolutional approaches that produce pixel-wise outputs, the hourglass network pools down to a very low resolution, then upsamples and combines features across multiple resolutions [15, 23]. On the other hand, the hourglass differs from prior designs primarily in its more symmetric topology.

We expand on a single hourglass by consecutively placing multiple hourglass modules together end-to-end. This allows for repeated bottom-up, top-down inference across scales. In conjunction with the use of intermediate supervision, repeated bidirectional inference is critical to the network’s final performance. The final network architecture achieves a significant improvement on the state-of-the-art for two standard pose estimation benchmarks (FLIC [1] and MPII Human Pose [21]). On MPII there is over a 2% average accuracy improvement across all joints, with as much as a 4-5% improvement on more difficult joints like the knees and ankles.¹



2 Related Work

With the introduction of “DeepPose” by Toshev et al. [24], research on human pose estimation began the shift from classic approaches [1–9] to deep networks. Toshev et al. use their network to directly regress the x,y coordinates of joints. The work by Tompson et al. [15] instead generates heatmaps by running an image through multiple resolution banks in parallel to simultaneously capture features at a variety of scales. Our network design largely builds off of their work, exploring how to capture information across scales and adapting their method for combining features across different resolutions.

¹ Code is available at <http://www-personal.umich.edu/~alnewell/pose>

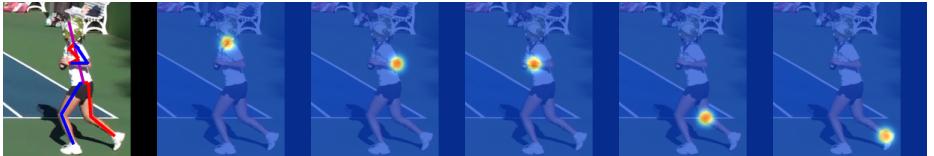


Fig. 2. Example output produced by our network. On the left we see the final pose estimate provided by the max activations across each heatmap. On the right we show sample heatmaps. (From left to right: neck, left elbow, left wrist, right knee, right ankle)

A critical feature of the method proposed by Tompson et al. [15] is the joint use of a ConvNet and a graphical model. Their graphical model learns typical spatial relationships between joints. Others have recently tackled this in similar ways [17, 20, 25] with variations on how to approach unary score generation and pairwise comparison of adjacent joints. Chen et al. [25] cluster detections into typical orientations so that when their classifier makes predictions additional information is available indicating the likely location of a neighboring joint. We achieve superior performance without the use of a graphical model or any explicit modeling of the human body.

There are several examples of methods making successive predictions for pose estimation. Carreira et al. [19] use what they refer to as Iterative Error Feedback. A set of predictions is included with the input, and each pass through the network further refines these predictions. Their method requires multi-stage training and the weights are shared across each iteration. Wei et al. [18] build on the work of multi-stage pose machines [26] but now with the use of ConvNets for feature extraction. Given our use of intermediate supervision, our work is similar in spirit to these methods, but our building block (the hourglass module) is different. Hu & Ramanan [27] have an architecture more similar to ours that can also be used for multiple stages of predictions, but their model ties weights in the bottom-up and top-down portions of computation as well as across iterations.

Tompson et al. build on their work in [15] with a cascade to refine predictions. This serves to increase efficiency and reduce memory usage of their method while improving localization performance in the high precision range [16]. One consideration is that for many failure cases a refinement of position within a local window would not offer much improvement since error cases often consist of either occluded or misattributed limbs. For both situations, any further evaluation at a local scale will not improve the prediction.

There are variations to the pose estimation problem which include the use of additional features such as depth or motion cues. [28–30] Also, there is the more challenging task of simultaneous annotation of multiple people [17, 31]. In addition, there is work like that of Oliveira et al. [32] that performs human part segmentation based on fully convolutional networks [23]. Our work focuses solely on the task of keypoint localization of a single person’s pose from an RGB image.

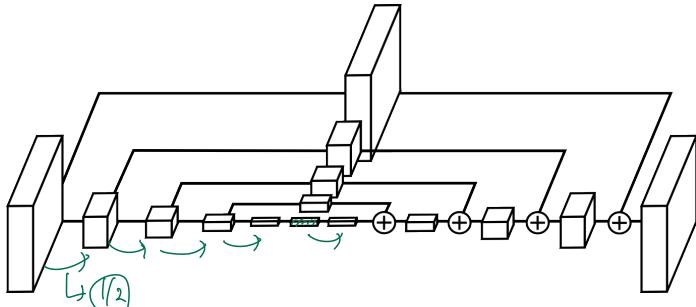


Fig. 3. An illustration of a single “hourglass” module. Each box in the figure corresponds to a residual module as seen in Figure 4. The number of features is consistent across the whole hourglass.

Our hourglass module before stacking is closely connected to fully convolutional networks [23] and other designs that process spatial information at multiple scales for dense prediction [15, 33–41]. Xie et al. [33] give a summary of typical architectures. Our hourglass module differs from these designs mainly in its more symmetric distribution of capacity between bottom-up processing (from high resolutions to low resolutions) and top-down processing (from low resolutions to high resolutions). For example, fully convolutional networks [23] and holistically-nested architectures [33] are both heavy in bottom-up processing but light in their top-down processing, which consists only of a (weighted) merging of predictions across multiple scales. Fully convolutional networks are also trained in multiple stages.

The hourglass module before stacking is also related to conv-deconv and encoder-decoder architectures [42–45]. Noh et al. [42] use the conv-deconv architecture to do semantic segmentation, Rematas et al. [44] use it to predict reflectance maps of objects. Zhao et al. [43] develop a unified framework for supervised, unsupervised and semi-supervised learning by adding a reconstruction loss. Yang et al. [46] employ an encoder-decoder architecture without skip connections for image generation. Rasmus et al. [47] propose a denoising auto-encoder with special, “modulated” skip connections for unsupervised/semi-supervised feature learning. The symmetric topology of these networks is similar, but the nature of the operations is quite different in that we do not use unpooling or deconv layers. Instead, we rely on simple nearest neighbor upsampling and skip connections for top-down processing. Another major difference of our work is that we perform repeated bottom-up, top-down inference by stacking multiple hourglasses.

3 Network Architecture

3.1 Hourglass Design

The design of the hourglass is motivated by the need to capture information at every scale. While local evidence is essential for identifying features like faces and

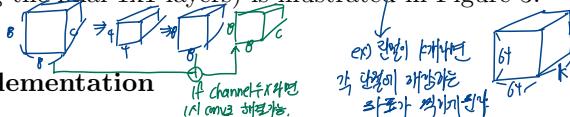
hands, a final pose estimate requires a coherent understanding of the full body. The person's orientation, the arrangement of their limbs, and the relationships of adjacent joints are among the many cues that are best recognized at different scales in the image. The hourglass is a simple, minimal design that has the capacity to capture all of these features and bring them together to output pixel-wise predictions.

즉, 관절 하나하나 씩은 저수준의 특징맵을 필요로하고, 관절과 관절사이의 관계 혹은 패턴을 파악 / 하기 위해서는 고수준의 이미지가 필요로 한다.

The network must have some mechanism to effectively process and consolidate features across scales. Some approaches tackle this with the use of separate pipelines that process the image independently at multiple resolutions and combine features later on in the network [15, 18]. Instead, we choose to **use a single pipeline** with **skip layers to preserve spatial information** at each resolution. The network reaches its lowest resolution at **4x4 pixels** allowing smaller spatial filters to be applied that compare features across the entire space of the image.

The hourglass is set up as follows: Convolutional and max pooling layers are used to process features down to a very low resolution. At each max pooling step, the network branches off and applies more convolutions at the original pre-pooled resolution. After reaching the lowest resolution, the network begins the top-down sequence of upsampling and combination of features across scales. To bring together information across two adjacent resolutions, we follow the process described by Tompson et al. [15] and do nearest neighbor upsampling of the lower resolution followed by an elementwise addition of the two sets of features. The topology of the hourglass is symmetric, so for every layer present on the way down there is a corresponding layer going up.

After reaching the output resolution of the network, two consecutive rounds of **1x1 convolutions** are applied to produce the final network predictions. The output of the network is a set of heatmaps where for a given heatmap the network predicts the probability of a joint's presence at each and every pixel. The full module (excluding the final 1x1 layers) is illustrated in Figure 3.



3.2 Layer Implementation

While maintaining the overall hourglass shape, there is still some flexibility in the specific implementation of layers. Different choices can have a moderate impact on the final performance and training of the network. We explore several options for layer design in our network. Recent work has shown the value of reduction steps with 1x1 convolutions, as well as the benefits of using consecutive smaller filters to capture a larger spatial context. [12, 14] For example, one can replace a 5x5 filter with two separate 3x3 filters. We tested our overall network design, swapping in different layer modules based off of these insights. We experienced an increase in network performance after switching from standard convolutional layers with large filters and no reduction steps to newer methods like the residual learning modules presented by He et al. [14] and “Inception”-based designs [12]. After the initial performance improvement with these types of designs, various additional explorations and modifications to the layers did little to further boost performance or training time.

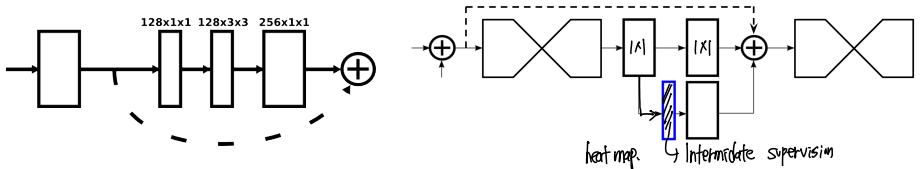


Fig. 4. Left: Residual Module [14] that we use throughout our network. **Right:** Illustration of the intermediate supervision process. The network splits and produces a set of heatmaps (outlined in blue) where a loss can be applied. A 1x1 convolution remaps the heatmaps to match the number of channels of the intermediate features. These are added together along with the features from the preceding hourglass.

Our final design makes extensive use of residual modules. Filters greater than 3x3 are never used, and the bottlenecking restricts the total number of parameters at each layer curtailing total memory usage. The module used in our network is shown in Figure 4. To put this into the context of the full network design, each box in Figure 3 represents a single residual module.

Operating at the full input resolution of 256x256 requires a significant amount of GPU memory, so the highest resolution of the hourglass (and thus the final output resolution) is 64x64. This does not affect the network’s ability to produce precise joint predictions. The full network starts with a 7x7 convolutional layer with stride 2, followed by a residual module and a round of max pooling to bring the resolution down from 256 to 64. Two subsequent residual modules precede the hourglass shown in Figure 3. Across the entire hourglass all residual modules output 256 features.

3.3 Stacked Hourglass with Intermediate Supervision

We take our network architecture further by stacking multiple hourglasses end-to-end, feeding the output of one as input into the next. This provides the network with a mechanism for repeated bottom-up, top-down inference allowing for reevaluation of initial estimates and features across the whole image. The key to this approach is the prediction of intermediate heatmaps upon which we can apply a loss. Predictions are generated after passing through each hourglass where the network has had an opportunity to process features at both local and global contexts. Subsequent hourglass modules allow these high level features to be processed again to further evaluate and reassess higher order spatial relationships. This is similar to other pose estimations methods that have demonstrated strong performance with multiple iterative stages and intermediate supervision [18, 19, 30].

Consider the limits of applying intermediate supervision with only the use of a single hourglass module. What would be an appropriate place in the pipeline to generate an initial set of predictions? Most higher order features are present only at lower resolutions except at the very end when upsampling occurs. If supervision is provided after the network does upsampling then there is no way for these features to be reevaluated relative to each other in a larger global

context. If we want the network to best refine predictions, these predictions cannot be exclusively evaluated at a local scale. The relationship to other joint predictions as well as the general context and understanding of the full image is crucial. Applying supervision earlier in the pipeline before pooling is a possibility, but at this point the features at a given pixel are the result of processing a relatively local receptive field and are thus ignorant of critical global cues.

Repeated bottom-up, top-down inference with stacked hourglasses alleviates these concerns. Local and global cues are integrated within each hourglass module, and asking the network to produce early predictions requires it to have a high-level understanding of the image while only partway through the full network. Subsequent stages of bottom-up, top-down processing allow for a deeper reconsideration of these features.

This approach for going back and forth between scales is particularly important because preserving the spatial location of features is essential to do the final localization step. The precise position of a joint is an indispensable cue for other decisions being made by the network. With a structured problem like pose estimation, the output is an interplay of many different features that should come together to form a coherent understanding of the scene. Contradicting evidence and anatomic impossibility are big giveaways that somewhere along the line a mistake was made, and by going back and forth the network can maintain precise local information while considering and then reconsidering the overall coherence of the features.

We reintegrate intermediate predictions back into the feature space by mapping them to a larger number of channels with an additional 1×1 convolution. These are added back to the intermediate features from the hourglass along with the features output from the previous hourglass stage (visualized in Figure 4). The resulting output serves directly as the input for the following hourglass module which generates another set of predictions. In the final network design, eight hourglasses are used. It is important to note that weights are not shared across hourglass modules, and a loss is applied to the predictions of all hourglasses using the same ground truth. The details for the loss and ground truth are described below.

3.4 Training Details

We evaluate our network on two benchmark datasets, FLIC [1] and MPII Human Pose [21]. FLIC is composed of 5003 images (3987 training, 1016 testing) taken from films. The images are annotated on the upper body with most figures facing the camera straight on. MPII Human Pose consists of around 25k images with annotations for multiple people providing 40k annotated samples (28k training, 11k testing). The test annotations are not provided so in all of our experiments we train on a subset of training images while evaluating on a heldout validation set of around 3000 samples. MPII consists of images taken from a wide range of human activities with a challenging array of widely articulated full-body poses.

There are often multiple people visible in a given input image, but without a graphical model or other postprocessing step the image must convey all necessary

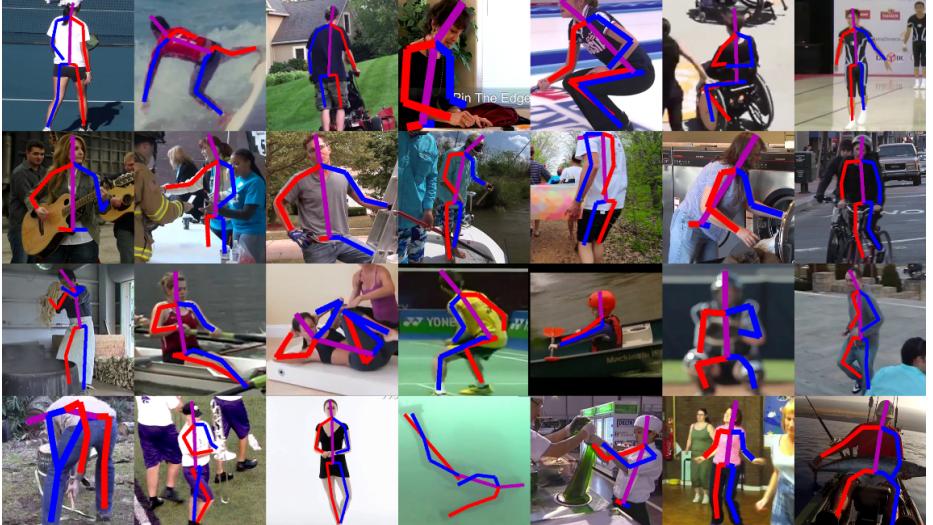
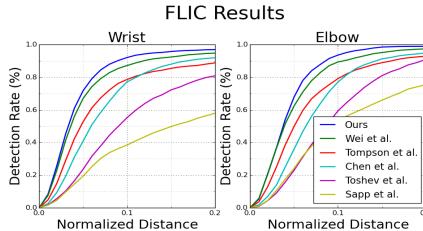


Fig. 5. Example output on MPII’s test set.

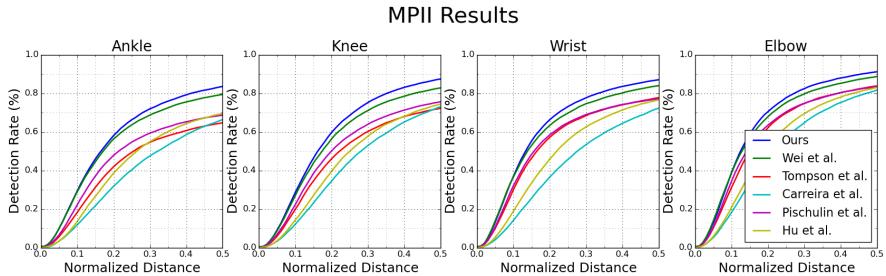
information for the network to determine which person deserves the annotation. We deal with this by training the network to exclusively annotate the person in the direct center. This is done in FLIC by centering along the x-axis according to the torsobox annotation - no vertical adjustment or scale normalization is done. For MPII, it is standard to utilize the scale and center annotations provided with all images. For each sample, these values are used to crop the image around the target person. All input images are then resized to 256x256 pixels. We do data augmentation that includes rotation ($+/- 30$ degrees), and scaling (.75-1.25). We avoid translation augmentation of the image since location of the target person is the critical cue determining who should be annotated by the network.

The network is trained using Torch7 [48] and for optimization we use rmsprop [49] with a learning rate of 2.5e-4. Training takes about 3 days on a 12 GB NVIDIA TitanX GPU. We drop the learning rate once by a factor of 5 after validation accuracy plateaus. Batch normalization [13] is also used to improve training. A single forward pass of the network takes 75 ms. For generating final test predictions we run both the original input and a flipped version of the image through the network and average the heatmaps together (accounting for a 1% average improvement on validation). The final prediction of the network is the max activating location of the heatmap for a given joint.

The same technique as Tompson et al. [15] is used for supervision. A Mean-Squared Error (MSE) loss is applied comparing the predicted heatmap to a ground-truth heatmap consisting of a 2D gaussian (with standard deviation of 1 px) centered on the joint location. To improve performance at high precision thresholds the prediction is offset by a quarter of a pixel in the direction of its next highest neighbor before transforming back to the original coordinate space of the image. In MPII Human Pose, some joints do not have a corresponding

**Fig. 6.** PCK comparison on FLIC

	Elbow	Wrist
Sapp et al. [1]	76.5	59.1
Toshev et al. [24]	92.3	82.0
Tompson et al. [16]	93.1	89.0
Chen et al. [25]	95.3	92.4
Wei et al. [18]	97.6	95.0
Our model	99.0	97.0

Table 1. FLIC results (PCK@0.2)**Fig. 7.** PCKh comparison on MPII

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Tompson et al. [16], CVPR'15	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Carreira et al. [19], CVPR'16	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Pishchulin et al. [17], CVPR'16	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Hu et al. [27], CVPR'16	95.0	91.6	83.0	76.6	81.9	74.5	69.5	82.4
Wei et al. [18], CVPR'16	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Our model	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9

Table 2. Results on MPII Human Pose (PCKh@0.5)

ground truth annotation. In these cases the joint is either truncated or severely occluded, so for supervision a ground truth heatmap of all zeros is provided.

4 Results

4.1 Evaluation

Evaluation is done using the standard Percentage of Correct Keypoints (PCK) metric which reports the percentage of detections that fall within a normalized distance of the ground truth. For FLIC, distance is normalized by torso size, and for MPII, by a fraction of the head size (referred to as PCKh).

FLIC: Results can be seen in Figure 6 and Table 1. Our results on FLIC are very competitive reaching 99% PCK@0.2 accuracy on the elbow, and 97% on the wrist. It is important to note that these results are observer-centric, which is consistent with how others have evaluated their output on FLIC.

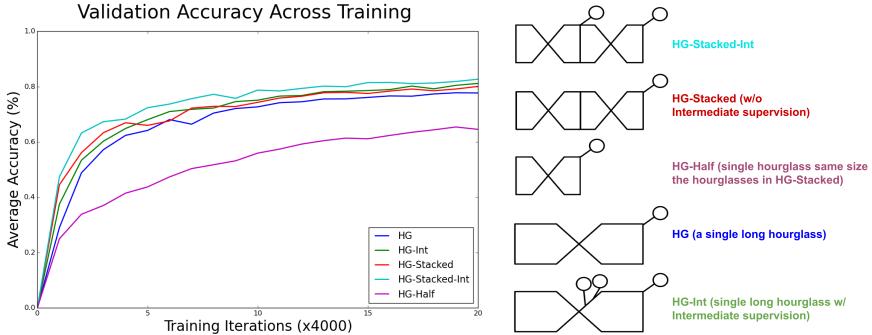


Fig. 8. Comparison of validation accuracy as training progresses. The accuracy is averaged across the wrists, elbows, knees, and ankles. The different network designs are illustrated on the right, the circle is used to indicate where a loss is applied

MPII: We achieve state-of-the-art results across all joints on the MPII Human Pose dataset. All numbers can be seen in Table 2 along with PCK curves in Figure 7. On difficult joints like the wrist, elbows, knees, and ankles we improve upon the most recent state-of-the-art results by an average of 3.5% (PCKh@0.5) with an average error rate of 12.8% down from 16.3%. The final elbow accuracy is 91.2% and wrist accuracy is 87.1%. Example predictions made by the network on MPII can be seen in Figure 5.

4.2 Ablation Experiments

We explore two main design choices in this work: the effect of stacking hourglass modules together, and the impact of intermediate supervision. These are not mutually independent as we are limited in how we can apply intermediate supervision depending on the overall architectural design. Applied separately, each has a positive impact on performance, and together we see a further improvements to training speed and in the end, final pose estimation performance. We look at the rate of training of a few different network designs. The results of which can be seen in Figure 8 which shows average accuracy on the validation set as training progresses. The accuracy metric considers all joints excluding those associated with the head and torso to allow for easier differentiation across experiments.

First, to explore the effect of the stacked hourglass design we must demonstrate that the change in performance is a function of the architecture shape and not attributed to an increase in capacity with a larger, deeper network. To make this comparison, we work from a baseline network consisting of eight hourglass modules stacked together. Each hourglass has a single residual module at each resolution as in Figure 3. We can shuffle these layers around for various network arrangements. A decrease in the number of hourglasses would result in an increase in the capacity of each hourglass. For example, a corresponding network could stack four hourglasses and have two consecutive residual modules at each

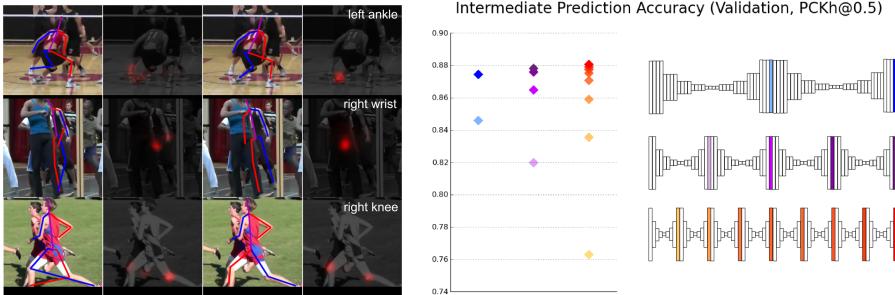


Fig. 9. **Left:** Example validation images illustrating the change in predictions from an intermediate stage (second hourglass) (left) to final predictions (eighth hourglass) (right). **Right:** Validation accuracy at intermediate stages of the network compared across different stacking arrangements.

resolution (or two hourglasses and four residual modules). This is illustrated in Figure 9. All networks share the same number of parameters and layers, though a slight difference is introduced when more intermediate supervision is applied.

To see the effect of these choices we first compare a two-stacked network with four residual modules at each stage in the hourglass, and a single hourglass but with eight residual modules instead. In Figure 8 these are referred to as HG-Stacked and HG respectively. A modest improvement in training can be seen when using the stacked design despite having approximately the same number of layers and parameters. Next, we consider the impact of intermediate supervision. For the two-stack network we follow the procedure described in the paper to apply supervision. Applying this same idea with a single hourglass is nontrivial since higher order global features are present only at lower resolutions, and the features across scales are not combined until late in the pipeline. We explore applying supervision at various points in the network, for example either before or after pooling and at various resolutions. The best performing method is shown as HG-Int in Figure 8 with intermediate supervision applied after up-sampling at the next two highest resolutions before the final output resolution. This supervision does offer an improvement to performance, but not enough to surpass the improvement when stacking is included (HG-Stacked-Int).

In Figure 9 we compare the validation accuracy of 2-, 4-, and 8-stack models that share approximately the same number of parameters, and include the accuracy of their intermediate predictions. There is a modest improvement in final performance for each successive increase in stacking from 87.4% to 87.8% to 88.1%. The effect is more notable at intermediate stages. For example, halfway through each network the corresponding accuracies of the intermediate predictions are: 84.6%, 86.5%, and 87.1%. Note that the accuracy halfway through the 8-stack network is just short of the final accuracy of the 2-stack network.

It is interesting to observe the mistakes made early and corrected later on by the network. A few examples are visualized in Figure 9. Common mistakes show up like a mix up of other people’s joints, or misattribution of left and



Fig. 10. The difference made by a slight translation and change of scale of the input image. The network determines who to generate an annotation for based on the central figure. The scaling and shift right of the input image is enough for the network to switch its predictions.

right. For the running figure, it is apparent from the final heatmap that the decision between left and right is still a bit ambiguous for the network. Given the appearance of the image, the confusion is justified. One case worth noting is the middle example where the network initially activates on the visible wrists in the image. Upon further processing the heatmap does not activate at all on the original locations, instead choosing a reasonable position for the occluded wrist.

5 Further Analysis

5.1 Multiple People

The issue of coherence becomes especially important when there are multiple people in an image. The network has to decide who to annotate, but there are limited options for communicating who exactly deserves the annotation. For the purposes of this work, the only signal provided is the centering and scaling of the target person trusting that the input will be clear enough to parse. Unfortunately, this occasionally leads to ambiguous situations when people are very close together or even overlapping as seen in Figure 10. Since we are training a system to generate pose predictions for a single person, the ideal output in an ambiguous situation would demonstrate a commitment to the joints of just one figure. Even if the predictions are lower quality, this would show a deeper understanding of the task at hand. Estimating a location for the wrist with a disregard for whom the wrist may belong is not desired behavior from a pose estimation system.

The results in Figure 10 are from an MPII test image. The network must produce predictions for both the boy and girl, and to do so, their respective center and scale annotations are provided. Using those values to crop input images for the network result in the first and third images of the figure. The center annotations for the two dancers are off by just 26 pixels in a 720x1280 image. Qualitatively, the most perceptible difference between the two input images is the change in scale. This difference is sufficient for the network to change its estimate entirely and predict the annotations for the correct figure.

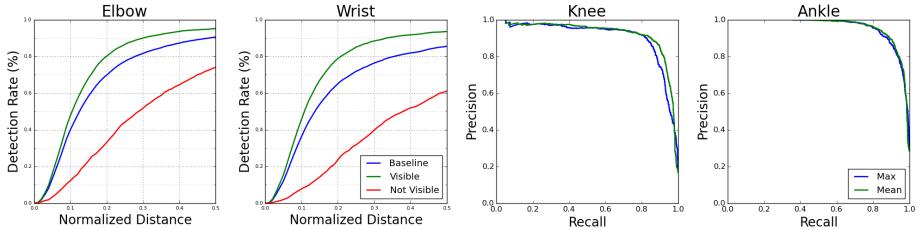


Fig. 11. **Left:** PCKh curves on validation comparing performance when exclusively considering joints that are visible (or not). **Right:** Precision recall curves showing the accuracy of predicting whether an annotation is present for a joint when thresholding on either the mean or max activation of a heatmap.

A more comprehensive management of annotations for multiple people is out of the scope of this work. Many of the system’s failure cases are a result of confusing the joints of multiple people, but it is promising that in many examples with severe overlap of figures the network will appropriately pick out a single figure to annotate.

5.2 Occlusion

Occlusion performance can be difficult to assess as it often falls into two distinct categories. The first consists of cases where a joint is not visible but its position is apparent given the context of the image. MPII generally provides ground truth locations for these joints, and an additional annotation indicates their lack of visibility. The second situation, on the other hand, occurs when there is absolutely no information about where a particular joint might be. For example, images where only the upper half of the person’s body is visible. In MPII these joints will not have a ground truth annotation associated with them.

Our system makes no use of the additional visibility annotations, but we can still take a look at the impact of visibility on performance. About 75% of the elbows and wrists with annotations are labeled visible in our held-out validation set. In Figure 11, we compare performance averaged across the whole validation set with performance on the three-quarters of joints that are visible and performance on the remaining quarter that are not. While only considering visible joints, wrist accuracy goes up to 93.6% from 85.5% (validation performance is slightly worse than test set performance of 87.1%). On the other hand, performance on exclusively occluded joints is 61.1%. For the elbow, accuracy goes from a baseline of 90.5% to 95.1% for visible joints and down to 74.0% for occluded joints. Occlusion is clearly a significant challenge, but the network still makes strong estimates in most cases. In many examples, the network prediction and ground-truth annotation may not agree while both residing in valid locations, and the ambiguity of the image means there is no way to determine which one is truly correct.

We also consider the more extreme case where a joint may be severely occluded or truncated and therefore have no annotation at all. The PCK metric used when evaluating pose estimation systems does not reflect how well these situations are recognized by the network. If there is no ground truth annotation provided for a joint it is impossible to assess the quality of the prediction made by the system, so it is not counted towards the final reported PCK value. Because of this, there is no harm in generating predictions for all joints even though the predictions for completely occluded or truncated joints will make no sense. For use in a real system, a degree of metaknowledge is essential, and the understanding that no good prediction can be made on a particular joint is very important. We observe that our network gives consistent and accurate predictions of whether or not a ground truth annotation is available for a joint.

We consider the ankle and knee for this analysis since these are occluded most often. Lower limbs are frequently cropped from images, and if we were to always visualize all joint predictions of our network, example pose figures would look unacceptable given the nonsensical lower body predictions made in these situations. For a simple way to filter out these cases we examine how well one can determine the presence of an annotation for a joint given the corresponding heatmap activation. We consider thresholding on either the maximum value of the heatmap or its mean. The corresponding precision-recall curves can be seen in Figure 11. We find that based solely off of the mean activation of a heatmap it is possible to correctly assess the presence of an annotation for the knee with an AUC of 92.1% and an annotation for the ankle with an AUC of 96.0%. This was done on a validation set of 2958 samples of which 16.1% of possible knees and 28.4% of possible ankles do not have a ground truth annotation. This is a promising result demonstrating that the heatmap serves as a useful signal indicating cases of truncation and severe occlusion in images.

6 Conclusion

We demonstrate the effectiveness of a stacked hourglass network for producing human pose estimates. The network handles a diverse and challenging set of poses with a simple mechanism for reevaluation and assessment of initial predictions. Intermediate supervision is critical for training the network, working best in the context of stacked hourglass modules. There still exist difficult cases not handled perfectly by the network, but overall our system shows robust performance to a variety of challenges including heavy occlusion and multiple people in close proximity.

References

1. Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 3674–3681

2. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
3. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong appearance and expressive spatial models for human pose estimation. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 3487–3494
4. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 1365–1372
5. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1465–1472
6. Ramanan, D.: Learning to parse images of articulated objects. Advances in Neural Information Processing Systems **134** (2006)
7. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(12) (2013) 2878–2890
8. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
9. Ladicky, L., Torr, P.H., Zisserman, A.: Human pose estimation using a joint pixel-wise and part-wise formulation. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 3578–3585
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
13. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning (2015)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Computer Vision and Pattern Recognition, 2016. CVPR 2016. IEEE Conference on (2015)
15. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems. (2014) 1799–1807
16. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 648–656
17. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2015)
18. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2016)

19. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2016)
20. Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2015) 1347–1355
21. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 3686–3693
22. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: Proceedings of the British Machine Vision Conference. (2010) doi:10.5244/C.24.12.
23. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
24. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 1653–1660
25. Chen, X., Yuille, A.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: Advances in Neural Information Processing Systems (NIPS). (2014)
26. Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J.A., Sheikh, Y.: Pose machines: Articulated pose estimation via inference machines. In: Computer Vision–ECCV 2014. Springer (2014) 33–47
27. Hu, P., Ramanan, D.: Bottom-up and top-down reasoning with hierarchical rectified gaussians. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, IEEE (2016)
28. Jain, A., Tompson, J., LeCun, Y., Bregler, C.: Modeep: A deep learning framework using motion features for human pose estimation. In: Computer Vision–ACCV 2014. Springer (2014) 302–315
29. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56**(1) (2013) 116–124
30. Pfister, T., Charles, J., Zisserman, A.: Flowing convnets for human pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1913–1921
31. Chen, X., Yuille, A.L.: Parsing occluded people by flexible compositions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3945–3954
32. Oliveira, G.L., Valada, A., Bollen, C., Burgard, W., Brox, T.: Deep learning for human part discovery in images. In: IEEE International Conference on Robotics and Automation (ICRA). (2016)
33. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1395–1403
34. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. (2014) 2366–2374
35. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(8) (2013) 1915–1929

36. Pinheiro, P., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). (2014) 82–90
37. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658
38. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. International Conference on Learning Representations (ICLR) (2016)
39. Couprie, C., Farabet, C., Najman, L., LeCun, Y.: Indoor semantic segmentation using depth information. International Conference on Learning Representations (ICLR) (2013)
40. Bertasius, G., Shi, J., Torresani, L.: Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4380–4389
41. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 447–456
42. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1520–1528
43. Zhao, J., Mathieu, M., Goroshin, R., Lecun, Y.: Stacked what-where auto-encoders. arXiv preprint arXiv:1506.02351 (2015)
44. Rematas, K., Ritschel, T., Fritz, M., Gavves, E., Tuytelaars, T.: Deep reflectance maps. Computer Vision and Pattern Recognition, 2016. CVPR 2016. IEEE Conference on (2015)
45. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
46. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 1099–1107
47. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. (2015) 3546–3554
48. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop. (2011)
49. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012)