R-CNNs for Pose Estimation and Action Detection

Georgia Gkioxari
University of California, Berkeley
gkioxari@berkeley.edu

Bharath Hariharan
University of California, Berkeley
bharath2@eecs.berkeley.edu

Ross Girshick
University of California, Berkeley
rbg@berkeley.edu

Jitendra Malik
University of California, Berkeley
malik@berkeley.edu

Abstract

We present convolutional neural networks for the tasks of keypoint (pose) prediction and action classification of people in unconstrained images. Our approach involves training an R-CNN detector with loss functions depending on the task being tackled. We evaluate our method on the challenging PASCAL VOC dataset and compare it to previous leading approaches. Our method gives state-of-the-art results for keypoint and action prediction. Additionally, we introduce a new dataset for *action detection*, the task of simultaneously localizing people and classifying their actions, and present results using our approach.

1 Introduction

In this paper we investigate a deep learning approach to the problems of pose estimation and action classification. We build on the R-CNN [12] object detection framework by training convolutional neural networks (CNNs) for each task.

R-CNN is a *monolithic* approach, since it trains a one component detector. This is different from previous state-of-the-art methods in object detection, such as DPM [9], where multiple components and latent part models are used, or poselets [4, 14], where an ensemble of supervised part detectors are combined to make better object and keypoint predictions.

Like R-CNN, our method takes as input an object proposal. However, in addition to predicting the object class (person versus not person), our method also makes a confidence-scored prediction for the person's pose and action. For the task of pose estimation, our system outputs a set of keypoints, with scores, for each object proposal. These predictions are evaluated under a detection setting, by measuring the AP for each keypoint independently [26]. Our approach achieves a mean AP of 15.2% on the PASCAL VOC 2009 person detection val dataset[1], which is an improvement over the previous state-of-the-art [14] mean AP of 12.7%. For the task of action classification, for each proposal a score is associated with the action prediction. The evaluation measures AP for each action independently. Our approach achieves 70.5% mAP on the PASCAL VOC action test set for action classification and is slightly better than the previous leading method [20] (70.2%), which also uses CNNs.

The standard method for evaluating action classification (reported above) assumes that ground-truth object locations are given at test time and one only needs to output an action label. Knowledge of the ground truth at test time makes this task unrealistic for real-world applications. Therefore, we introduce a new task, which we call *action detection*. In action detection, ground truth is unavailable at test time, thus an approach is expected to make predictions of the location of the person and the action being performed. Evaluation follows the standard AP metric. Intuitively, one can think of this task as introducing new object categories defined by (person, action) pairs, and then applying

[1] This is the standard dataset for benchmarking human pose estimation in PASCAL [13]

the standard PASCAL VOC object detection evaluation [8] to these new categories. We consider several variants of R-CNN on this metric and compare their performance. The dataset is available online (`http://www.cs.berkeley.edu/~gkioxari/`) and we hope that other researchers will use it to evaluate their methods.

This report is organized as follows. In Section 2 we present relevant past work. Section 3 describes our task-specific loss functions, while in Section 4 we present in detail our experiments and results.

2 Related work

While CNNs ([18]) achieved state-of-the-art performance on handwritten digit classification, most recognition work has used more traditional pipelines based on hand-engineered features. However, recently Krizhevsky *et al.* [17] showed a big gain over existing methods in image classification on the ImageNet challenge [5]. This rekindled interest in CNNs. Donahue *et al.* [6] found that Krizhevsky *et al.*'s model can be used to extract features that are broadly useful for a large variety of tasks. Girshick *et al.* [12] found that using these features led to state-of-the-art object detection performance. They also found that "fine-tuning" the network (*i.e.*, continued optimization of the network on a new task and/or dataset after starting from the initial ImageNet model) led to even greater gains. These results show that all these computer vision tasks are indeed related, and indicate that further improvements may be achieved through multitask training. CNNs have also given improvements on action classification and pose estimation, although gains have been somewhat smaller. Oquab *et al.* [20] show small but significant improvements over state-of-the-art in action classification by finetuning a network pretrained on Imagenet. For pretraining, they add data from an additional 512 classes that they think are relevant. For pose estimation, Toshev *et al.* [24] learn a cascade of neural networks that regress to joint locations, with each successive layer refining the output of the last. At each stage, Toshev *et al.* augment data by randomly jittering input windows. The authors show improvements on many joints for multiple datasets. The gains mostly come in the low precision regime, indicating that the network does a good job of providing a rough location.

Previously leading, non neural network based approaches for the tasks of pose estimation used part detectors to provide accurate locations of the keypoints. Fischler and Elschlager [11] introduced pictorial structure models in 1973. Felzenszwalb and Huttenlocher [10] later rephrased PSM under a probabilistic framework. In their model, the body parts are represented as rectangular boxes and their geometric constraints are captured by tree-structured graphs. Ramanan [22], Andriluka *et al.* [2] and Eichner *et al.* [7] exploited appearance features to parse the human body. Johnson and Everingham [15] replaced a single PSM with a mixture of PSMs in order to capture more variety in pose. Yang and Ramanan [26] improved on pictorial structures mixtures by using a mixture of templates for each part. Sapp *et al.* [23] enhanced the appearance models, including contour and segmentation cues. Wang *et al.* [25] used a hierarchy of poselets for human parsing. Pishchulin *et al.* [21] used PSM with poselets as parts for the task of human body pose estimation. Gkioxari *et al.* [13] trained arm specific detectors and show state-of-the-art results on the PASCAL VOC dataset for arm pose prediction given ground truth location of the person, but did not propose an approach for whole-body pose estimation. Recently, Gkioxari *et al.* [14] introduced k-poselets, which are deformable part models composed of poselets, and showed state of the art performance on keypoint prediction for all keypoints on PASCAL VOC under a detection setting, where the location and number of the people in the image is unknown.

For the task of action classification from 2D images, most leading approaches are using part detectors to learn action specific classifiers. In detail, Maji *et al.* [19] train action specific poselets and for each instance create a poselet activation vector that is being classified using SVMs. Hoai *et al.* [1] use body-part detectors to localize the human and align feature descriptors and achieve state-of-the-art results on the PASCAL VOC 2012 action classification challenge. Khosla *et al.* [16] identify the image regions that distinguish different classes by sampling regions from a dense sampling space and using a random forest algorithm with discriminative classifiers.

3 A Single Convolutional Neural Network for Multiple Tasks

In this work, we use a single CNN that is trained jointly for multiple tasks. Figure 1 shows an example of a multitask CNN. Each task is associated with a loss function. We present loss functions for the tasks of person detection, pose estimation and action classification.

Person detection. Person detection is the task of predicting the location of people in an image. A detection is considered correct if the intersection over union of the predicted and ground truth bounding box is more than a threshold (usually 0.5). During R-CNN fine-tuning for the task of person detection, a region x is positive $(l = 1)$ if it overlaps more than 0.5 with a ground truth person in the image, and negative $(l = 0)$ if it overlaps less than 0.3. All the other regions are not considered. The output $y = [p_0, p_1]$ of the CNN is a two-dimensional probability vector, where p_1 indicates the probablity that x is a person and $p_0 = 1 - p_1$. The loss associated with the person detection task is a log loss

$$\text{loss}_D = -(1 - l) \log p_0 - l \log p_1 \tag{1}$$

Pose estimation. Pose estimation is the task of predicting the location of specific keypoints in the human body. A prediction is correct if the estimation of the location of the keypoints is within some distance from the ground truth location. During R-CNN fine-tuning for the task of pose estimation, a region x which overlaps more than 0.5 with an instance in the image is accompanied with a set of keypoint (x, y) locations and visibility flags, $\{(x_k, y_k, v_k)\}_{k=1}^{|K|}$, which belong to that instance. The keypoint locations are normalized with respect to the center and width and height of the region. The output of the R-CNN, $\{(\hat{x}_k, \hat{y}_k)\}_{k=1}^{|K|}$, is a set of predicted keypoint locations. The loss associated with the pose estimation task is the mean squared error between the prediction and the ground truth, ignoring keypoints which are not visible

$$\text{loss}_P = \frac{1}{|K|} \sum_{k=1}^{|K|} v_k \left((\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2 \right). \tag{2}$$

For regions which do not overlap sufficiently with any ground truth instance, the loss is 0.

Action classification. Action classification is the task of predicting the action a person is performing. A prediction is correct if the predicted action is correct. During R-CNN fine-tuning for this task, a region x is marked with an action label $l = \alpha$ if it overlaps more than 0.7 with a person performing action α. We consider a higher threshold for the overlap compared to pose to ensure that the regions being used at training time are significantly close to the ground truth. The output of the R-CNN, $y = (p_1, ..., p_{|A|})$, is a probability vector, where p_α indicates how likely x contains action α. The loss associated with this task is a log loss

$$\text{loss}_A = -\sum_{\alpha=1}^{|A|} [l = \alpha] \log p_\alpha, \tag{3}$$

where $[P]$ is the Iverson bracket notation for a boolean statement P. As for pose, regions which do not overlap sufficiently with ground truth always have a loss of 0.

If x is an example with true labels $L_x = \{l_i\}_{i=1}^N$ and $f(x) = \{y_i\}_{i=1}^N$ is the output of the network, where y_i is the output of the i-th task, the total loss of the network could be the weighted sum of the individual losses

$$\text{loss}(f(x), L_x) = \sum_{i=1}^{N} \lambda_i \text{loss}_i(y_i, l_i), \tag{4}$$

where loss_i is the loss function associated with the i-th task and the parameter λ_i is defined based on the importance of the task in the overall loss.

4 Experiments

In this section, we describe in detail our experiments using CNNs for the tasks of pose estimation and action prediction. We use the CNN architecture, as defined by [17], which has shown state of the art results for image classification on ImageNet. We build on the R-CNN work [12] which

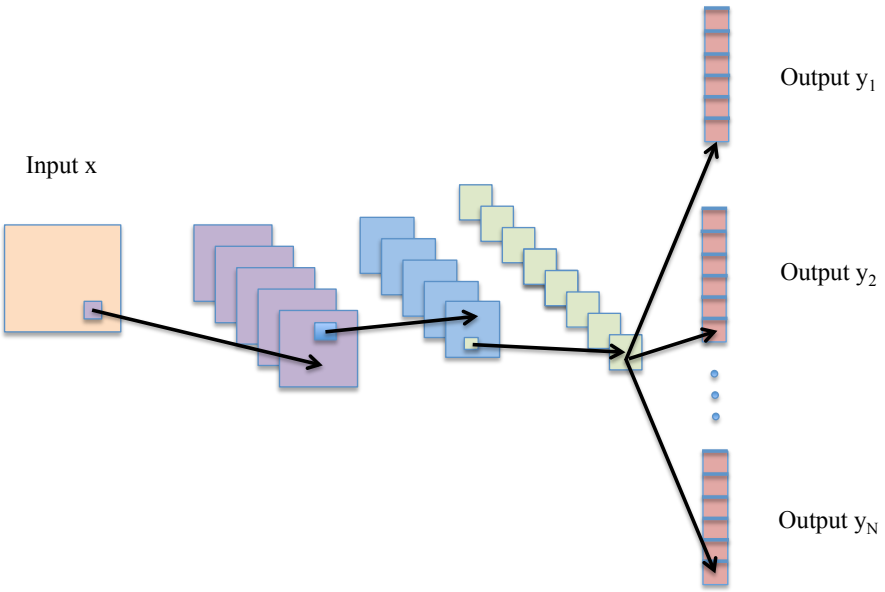

Figure 1: An example of a convolutional neural network. The network takes a single input and computes N outputs. Each output is associated with a different loss function. The biggest part of the network is shared, while task specific parts exist only close to the output of the network.

recently demonstrated an impressive improvement from previous state of the art approaches on object detection on the PASCAL VOC and ImageNet. We use region proposals generated by [3] and fine-tune the R-CNN starting from the ImageNet pretrained model on 1000 classes.

4.1 Datasets & Evaluation

Pose. For the pose estimation task we use the subset of PASCAL VOC 2012 detection dataset that has people, augmented with keypoint annotations. To train the R-CNN we use the *train* set. For parameter selection and evaluation, we split the person images in the PASCAL VOC 2009 *val* set into two sets: VAL09A is used for parameter selection, and VAL09B for test evaluation. The VAL09A subset consists of 723 images of 1529 instances and the VAL09B subset consists of 723 images of 1467 instances. For pose, VAL09A is used to train classifiers that estimate the confidence of the keypoint predictions produced by R-CNN. We evaluate the keypoint prediction using APK (average precision of keypoint), as defined in [26]. APK measures the correctness of the keypoint predictions under a detection setting, where each prediction is associated with a score and those are subsequently used to compute precision-recall curves. A keypoint prediction is correct if the distance from the ground truth is less than a $0.2 \cdot H$, where H is the height of the torso of the corresponding ground truth instance.

Action. For the action task we use the PASCAL VOC 2012 action dataset. To train the R-CNN we use the train set, which consists of 2296 images and 3134 instances (with their bounding box and action labels). There are in total 10 different actions. We use the PASCAL VOC 2012 test set to evaluate for the task of action classification. The action classification task assumes knowledge of the ground truth location of the people during test time. Therefore, not all person instances are necessarily annotated in the images. In order to evaluate for the task of action localization and classification, which we call *action detection*, we annotated all the people on the validation set with bounding boxes and actions. This increased the number of annotated instances from 3144 to 5891. We use the augmented val set to evaluate for the task of action detection. For action classification we use the evaluation criterion defined by the PASCAL VOC action task, which computes the AP on the ground truth test boxes. For action detection, we measure the AP on all region proposals in

APK [$\alpha = 0.2$]	Nose	R_Shoulder	R_Elbow	R_Wrist	L_Shoulder	L_Elbow	L_Wrist	R_Hip	R_Knee	R_Ankle	L_Hip	L_Knee	L_Ankle	*mAP*
k-poselets	42.9	27.1	12.2	3.4	27.3	11.8	2.8	**10.6**	**4.4**	3.8	**11.4**	**4.9**	3.2	12.7
Pose R-CNN	**52.0**	**32.5**	**16.6**	**5.9**	**32.1**	**14.6**	**5.6**	9.7	4.0	**4.6**	10.8	4.8	**4.8**	**15.2**

Table 1: APK average precision (%) on the second half of PASCAL VOC val 2009 person dataset (VAL09B). We report keypoint prediction AP for k-poselets [14]. Pose R-CNN is better for almost all keypoints.

Figure 2: Examples of keypoint predictions from Pose R-CNN on ground truth regions on VAL09B. Ground truth information is only used for visualizatin purposes. The Nose and Eyes predictions are connected in a triangle, while the keypoints of the arms and the legs are visualized using stick figures.

the image, where a detection is correct if the overlap is more than 0.5 with a ground truth instance and the action is correctly predicted for that instance.

4.2 Experimental Results

4.2.1 Pose Estimation

We train a network, which we call *Pose R-CNN*, with the loss function

$$\text{loss} = \lambda_D \text{loss}_D + \lambda_P \text{loss}_P + \lambda_A \text{loss}_A \tag{5}$$

with $\lambda_D = \lambda_A = 0$ and $\lambda_P = 1$ (so, loss $= \text{loss}_P$).

During test time, for each input region the network predicts the locations of all the keypoints. In order to get a confidence score for those predictions, we train a linear SVM classifier for each of the keypoints. The SVM classifier is trained using as positive examples the fc7 features of the regions that make a correct prediction for that keypoint (as defined in the APK metric), and as negative all other regions. We train $|K|$ such classifiers on the VAL09A set, where $|K|$ is the number of keypoints. For each of the $|K|$ keypoints, we measure the AP of that keypoint on VAL09B. Table 1 shows the results of Pose R-CNN. For comparison with the state of the art approach, we show the perfomance of k-poselets [14] on the same set of images. Note that Pose R-CNN outperforms k-poselets with a relative improvement of 19.7%.

Figure 2 shows examples of the predictions of Pose R-CNN on ground truth regions of VAL09B. (We only use ground truth regions for visualization purposes.)

4.2.2 Action Classification

We train a network, which we call *Action R-CNN*, with the loss function

$$\text{loss} = \lambda_D \text{loss}_D + \lambda_P \text{loss}_P + \lambda_A \text{loss}_A \tag{6}$$

with $\lambda_D = \lambda_P = 0$ and $\lambda_A = 1$ (so, loss $= \text{loss}_A$).

For the task of action classification, the location of the people performing the action is considered known. Since there is too little data to train a large CNN, we augment the training set with regions that overlap more than 0.7 with the ground truth regions. This leads to an increase of 100x of the training set.

AP	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photo	Using Computer	Walking	*mAP*
Stanford [1]	75.7	44.8	66.6	**44.4**	93.2	94.2	87.6	38.4	**70.6**	**75.6**	69.1
Oxford [1]	**77.0**	**50.4**	65.3	39.5	94.1	**95.9**	**87.7**	42.7	68.6	74.5	69.6
Oquab *et al.* [20]	74.8	46.0	75.6	45.3	93.5	95.0	86.5	49.3	66.7	69.5	70.2
Action R-CNN + fc6 SVM	76.2	47.4	**77.5**	42.2	**94.9**	94.3	87.0	**52.9**	66.5	66.5	**70.5**

Table 2: AP (%) on Test 2012 for action classification. We report numbers for leading approaches in the task, according to the PASCAL VOC leaderboard [1]. Action R-CNN is better on average.

AP	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photo	Using Computer	Walking	*mAP*
Action R-CNN + fc6 SVM	14.4	8.4	9.4	4.7	9.6	19.0	16.4	10.8	3.1	3.7	10.0
Detection R-CNN + fc6 SVM	19.5	9.1	11.9	7.5	9.9	20.1	24.4	5.3	1.8	9.3	11.9
Detection-Action R-CNN	**29.6**	**22.4**	**28.4**	**16.8**	**26.2**	**35.2**	**28.1**	**22.7**	**15.8**	**20.6**	**24.6**

Table 3: AP (%) on the augmented action val 2012 dataset. We report numbers of various types of R-CNN networks. The network trained jointly for detection and action (Detection-Action R-CNN) outperfoms the rest of the R-CNNs by a significant margin.

During test time, a ground truth region is being marked with a score for each action. This score can come directly from the softmax output of the CNN or by training action specific classifiers (linear SVMs) on top of fc6 or fc7 features. In practice, we observed that fc6 features work slightly better than the softmax output and fc7, perhaps due to overfitting of the network. In addition, in order to make use of context, which is very useful for such classification tasks, we rescore our predictions. In particular, we train linear SVM classifiers on feature vectors composed of: (1) the predicted score of the action from Action R-CNN, (2) the maximum action scores from Action R-CNN of all the other instances in the image, if any, and (3) the maximum scores of the objects {horse, bike, motorcycle, tv monitor} that overlap more than 0.1 with the ground truth region. The score of the objects were computed using a regular R-CNN trained on the 20 object categories of the PASCAL VOC detection challenge, similar to [12].

Table 2 shows the results of Action R-CNN. We also show leading state of the art approaches in this field, based on the PASCAL VOC 2012 leaderboard [1]. Note that [20] use a CNN, like us, but fine-tune their network by learning two additional fully connected layers. They get leverage from the objects involved and the context around the people performing the action by initializing their network with a model trained on ImageNet with 1512 categories, including horse, bike etc., and by using bigger regions centered at the ground truth location. Action R-CNN perfoms slightly better on average compared to the rest of the approaches.

4.2.3 Action Detection

The task of action classification assumes knowledge of the location of the people. This makes the task a lot easier, since it skips the difficult step of detection. A real application cannot assume perfect localization of the objects to be classified. On the other hand, action detection is the task of localizing and classifying the people as perfoming one of $|A|$ actions. A correct prediction is one that overlaps more than 0.5 with a ground truth instance and predicts the correct action label.

We train a network, which we call *Detection-Action R-CNN*, with the loss function

$$\text{loss} = \lambda_D \text{loss}_D + \lambda_P \text{loss}_P + \lambda_A \text{loss}_A \tag{7}$$

with $\lambda_P = 0$ and $\lambda_D = \lambda_A = 1$.

We use the PASCAL VOC 2012 detection and action train set. We use the softmax output of the network to make action predictions, since it proved to work better than fc6 or fc7 features with SVMs. Table 3 shows the perfomance of Action R-CNN and Detection-Action R-CNN, as well as the performance of *Detection R-CNN* trained solely for the task of detection ($\lambda_D = 1, \lambda_P = \lambda_A = 0$). The Detection R-CNN and Action R-CNN make action predictions by using action specific SVM classifiers, which are trained on fc6 features on the PASCAL VOC action train set, after cross validating across choices of features.

Figure 3 shows examples of the Detection-Action R-CNN. For every image, we show the top 3 activations along with their predicted action labels and scores. It is clear from the examples that the network prefers regions around the face to predict *phoning* and *taking photo*, while it prefers

Figure 3: Examples of action detections from the Detection-Action R-CNN on the PASCAL VOC 2012 action val set. (We encourage the reader to view this figure on a computer and zoom in.)

regions that include the whole person for actions such as *walking* and *jumping*. Also, activations for *playing instrument*, *riding bike* and *riding horse* seem to be driven by the presence of objects in the proximity of the region.

4.2.4 Person Detection

For the task of person detection, the Detection R-CNN with a SVM classifier on fc7 features achieves an AP of 54.8% on the PASCAL VOC 2012 detection test set. (Note that the original work of R-CNN [12] reports 53.2% with no bbox regression. The difference in performance is probably due to the different nature of the regions [3].) The performance of the Detection-Action R-CNN with a similar SVM classifier achieves 56.0% on the PASCAL VOC 2012 test set for the task of person detection. This suggests that networks that are jointly trained for detection and other tasks might be useful for the task of detection as well.

4.2.5 A single network for Detection, Pose and Action

We train a network for all three tasks jointly, which we call *Detection-Pose-Action* R-CNN, with loss function

$$loss = \lambda_D loss_D + \lambda_P loss_P + \lambda_A loss_A \tag{8}$$

where $\lambda_D = \lambda_P = 1$ and $\lambda_A = 2$. We choose a higher value for action classification to make sure that the task has a significant contribution to the total loss, since there is significantly fewer training data for action compared to detection and pose.

The joint network for the three tasks performs on average similar to the networks trained for specific tasks individually. Specifically, for the task of person detection, the joint network achieves 56.4% on the PASCAL VOC 2012 test set. For pose, the mean AP for keypoint prediction on the VAL09B set is 15.5%, while for action detection, the mean AP is 21.6%.

The joint network behaves very similar to the individual networks, while at the same time it is N-times faster. It involves finetuning a single network instead of N networks and processing the evaluation data once instead of N times.

Acknowledgments

This work was supported by the Intel Visual Computing Center, ONR SMARTS MURI N000140911051, ONR MURI N000141010933, a Google Research Grant and a Microsoft Research fellowship. The GPUs used in this research were generously donated by the NVIDIA Corporation.

References

[1] http://pascallin.ecs.soton.ac.uk/challenges/voc/voc2012/, 2012.

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

[3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition*, 2014.

[4] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010.

[5] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). `http://www.image-net.org/challenges/LSVRC/2012/`.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

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

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010.

[10] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005.

[11] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. on Computer*, 22(1), 1973.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR*, 2013.

[14] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014.

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

[16] A. Khosla, B. Yao, and L. Fei-Fei. Integrating randomization and discrimination for classifying human-object interaction activities. In *Human-Centered Social Media Analytics*. Springer, 2014.

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

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.

[19] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011.

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[21] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013.

[22] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006.

[23] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.

[24] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[25] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, 2011.

[26] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures-of-parts. *TPAMI*, 2012.