Human Pose Estimation and Activity Classification Using Convolutional Neural Networks

Amy Bearman
Stanford University
abearman@cs.stanford.edu

Catherine Dong
Stanford University
cdong@cs.stanford.edu

Abstract

In this paper, we investigate the problems of human pose estimation and activity classification using a deep learning approach. We constructed a CNN to address the regression problem of human joint location estimation, and achieved a PDJ score of about 60%. Furthermore, using weight initializations from an AlexNet trained to classify on ImageNet, we trained a deep convolutional neural network (CNN) to classify images of humans based on the activity the person is performing. For this task of activity classification, we achieved a classification accuracy of 80.51% across 20 activity categories. Using our relatively simple models, we were able to produce results for human pose estimation that beat those of local detectors, which demonstrates the power of CNNs in challenging visual recognition tasks.

1. Introduction

The problem of human pose estimation involves the identification of the location of keypoints of the body, which includes major body parts and joints. There are various applications associated with this problem, such as action classification and body movement prediction. The identification of body keypoints has proved to be a challenging problem due to small joints, occlusions, and the need to capture context [10].

Convolutional neural networks (CNNs) have had remarkable success recently on image classification and object localization problems. They are very similar to ordinary neural networks in that they are made up of neurons with learnable weights and biases. However, neural networks don't scale well to larger images. Each neuron in a layer is fully connected to all the neurons in the previous layer, so we quickly generate a huge number of parameters and end up overfitting on the training set. CNNs take advantage of the fact that the input consists of images, so they constrain the architecture in a more sensible way which vastly reduces the number of parameters.

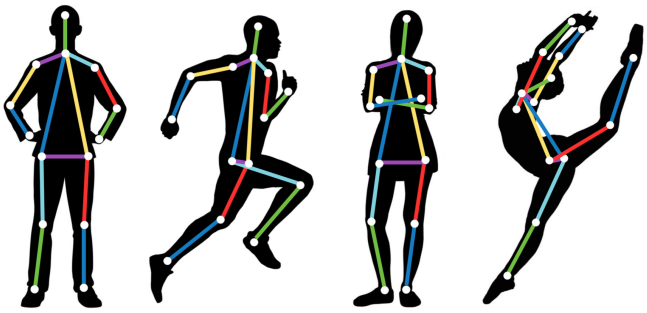

Figure 1. Example human pose trees.

CNNs are appealing for human pose estimation for two reasons. First, there's no need to explicitly design feature representations and detectors for parts, because a model and features are learned from the data. Second, the model learned is holistic, where the final joint estimates are based on a complex nonlinear transformation of the full image, as opposed to local detectors whose reasoning is constrained to a single part and can only model a small subset of interactions between body parts.

Here are just a few of the challenges in predicting human pose coordinates: the foreshortening of limbs, occlusion of limbs, rotation and orientation of the figure, and overlap of multiple subjects. Examples of especially challenging poses to annotate can be seen in Figure 2.

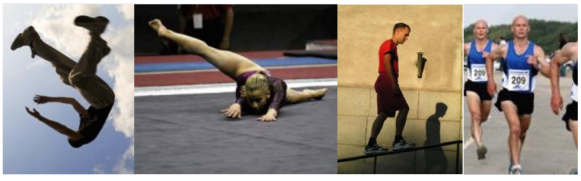

Figure 2. Difficult poses to annotate: rotation, foreshortening, occlusion, and multiple figures

This variability in the input form suggests that the holistic reasoning provided by CNNs may be a powerful strategy. In this project, we explore different CNN architectures for modeling human pose estimation and activity classification.

1.1. Problem Statement

Our problem consists of two tasks: human pose estimation and action classification. For pose estimation, our network takes as input a raw image and outputs a vector of coordinates of the body keypoints. We aim to identify x-y pixel coordinates for 14 body joints (as depicted in Figure 3). We train a regression CNN that minimizes loss as defined in **Section 3.2**.

For the latter problem, we aim to label the images based on activity category (20 classes, e.g. winter activity) and specific activity (410 classes, e.g. downhill skiing).

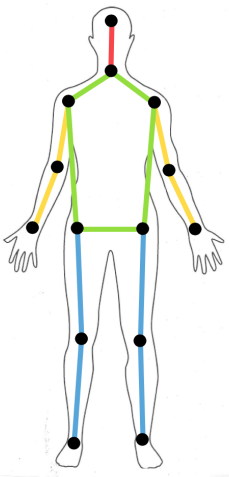

Figure 3. A depiction of the 14 joint keypoints we aim to identify.

2. Related Work

CNNs have historically been used for classification tasks, but they are increasingly being applied towards localization/detection problems. A classification CNN can be converted to a localization CNN by replacing the final classification layer with a regression layer for which the activations are real-valued predictions, and by using a regression loss function. Alternatively, Sermanet et al. proposes an integrated approach to object detection, recognition, and localization with a single CNN. [8].

At the high level, human activities can often be accurately characterized in terms of body pose, motion, and interaction with scene objects [7]. However, due to the challenging nature of this problem, most current activity recognition models rely on holistic representations that extract appearance and motion features from the video from which the images are pulled. Recently, Toshev et al. [10] showed that applying deep CNNs to pose estimation as a regression problem has the advantage of reasoning about pose in a simple but holistic fashion. They formulated pose estimation as a body joint regression problem, in which the location of each joint is predicted using a 7-layer CNN in which the input is the full image. This approach achieved a state-of-the-art PCP score (0.61) and was much simpler than previous methods based on explicitly designed feature representations and graphical models.

3. Approach

3.1. Pose Estimation/Joint Localization

Model Architecture

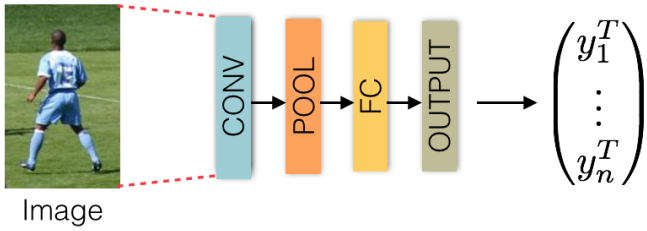

Figure 4. Illustration of the regression CNN. We omit repeated layer types.

We formulate the human pose estimation problem as a regression problem that can be modelled by a generic convolutional neural network. The CNN takes as input a full image (96×96 pixels) and outputs the pixel coordinates of each body keypoint. We used Lasagne (a library to build and train neural networks in Theano) to implement the regression net.

To express a pose, we encode the locations of all k body keypoints in a pose vector defined as $\mathbf{y} = \left((x^{(1)}, y^{(1)}), \ldots, (x^{(k)}, y^{(k)})\right)^T$. A labelled image in the training set is represented as $x, \mathbf{y})$, where x is the image data and $\mathbf{y}$ is the ground truth pose vector. The output of the CNN is a real-valued vector of 28 numbers representing the 14 concatenated (x, y) coordinates of the pose.

We use the Mean Squared Error (MSE) to represent the distance between our pose vector and the ground truth pose vector:

$$\text{MSE} = \frac{1}{N} \sum_{(x,\mathbf{y}) \in D} \sum_{i=1}^{k} (\mathbf{y}_i - f(w_i; x))^2 \qquad (1)$$

where N is the number of training examples, D is the training set, k is the number of body keypoints, and w_i is the weights learned for the i-th body keypoint.

Training Details

We use backpropagation to optimize for the weights w. We perform mini-batch gradient descent with Nesterov momentum over the training set with a batch size of 128. We vary the learning rate and momentum coefficient over time. The learning rate is initialized to 0.03 and terminated at 0.0001; momentum is initialized to 0.9 and terminated

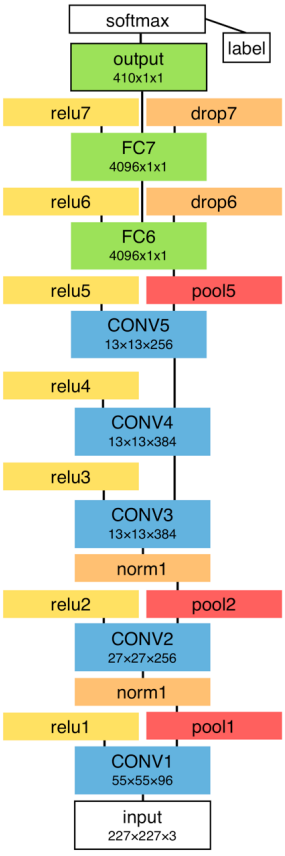

Figure 5. Classification CNN architecture for activity classification. The final layer outputs the probabilities of each of the 410 activity types.

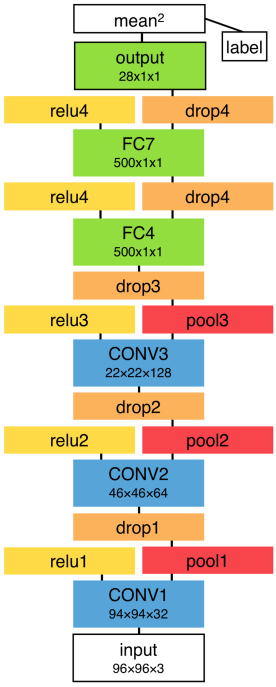

Figure 6. Regression CNN architecture for keypoint location estimation. The final layer outputs a 28-dimensional vector representing the x and y coordinates of each of the 14 keypoints.

at 0.999. This CNN was implemented with Theano and Lasagne [6]. See Figure 6 for a detailed figure with our regression architecture.

3.2. Activity Classification

Model Architecture

We formulate the activity classification problem as a multi-class classification problem that can be modelled by a convolutional neural network. The CNN takes as input a full image (256 $\times$ 256 pixels) and outputs a vector of numbers representing the probabilities of each of the activity labels for either the 410 specific activities or the 20 activity categories, depending on the ground truth lables passed in and the size of the final fully connected output layer.

This CNN is implemented using Caffe [3]. We use weight initializations from a pre-trained Caffe reference model based on AlexNet [5]. It consists of five convolution layers and three fully-connected layers interspersed with ReLU non-linearities, max-pooling, and normalization layers, with the final layer implementing the softmax function (Figure 5). The first convolutional layer has a depth of uses 11$\times$11 filters with a stride of 4, the second convolutional layer uses 5$\times$5 filters with a stride of 2, and the remaining three convolutional layers all use 3$\times$3 filters with a stride of 1. Furthermore, the max-pooling layers uses 3$\times$3 filters with a stride of 2.

Training Details

We train the CNN using 15,000 images from the MPII dataset labeled with the activity type and validated against 3,000 images. We train for 5,000 iterations using a batch size of 256. The base learning rate is 0.001, which we decrease step-wise by a factor of 0.1 every 1,000 iterations. We use a momentum of 0.9.

4. Setup

4.1. Data

The dataset we use for pose estimation is the Leeds Sports Pose Dataset (see Figure 7) and its extension. Together, they contain 11,000 training images and 1,000 test images. All test images are taken from the Leeds Sport Pose Dataset. The dataset contains images gathered from Flickr searches for "parkour," "gymnastics," and "athletics," which have been deemed difficult to annotate. The images have been scaled such that the most prominent person is roughly 150 pixels in length. All images are annotated with 14 body keypoints: right and left ankles, knees, hips, wrists, elbows, shoulders, and neck and top of head. Each body keypoint is an (x, y) pixel coordinate pair in the image space.

Figure 7. A sample of the Leeds Sports Pose dataset used for training the regression CNN.

The dataset we used for activity classification is the MPII Human Pose Dataset 8. This dataset contains approximately 25,000 images with over 40,000 people. Each image is extracted from a YouTube video, and all images are about 1280 $\times$ 720 pixels in size. The dataset covers 410 specific categories of human activity and 20 general categories. The images are each labeled with one of 410 activity IDs and one of 20 activity category labels. In addition, each body in the image is annotated with the bounding boxes of the body and head, the x-y coordinates of each keypoint joint, and an indication of whether or not the joint is visible. We used this dataset for training the activity classification

Figure 8. A sample of the MPII Human Pose dataset used for training the classification CNN. The individual images are labeled with the a specific activity type (410 classes total), and the images of each column are all part of the larger activity categories (20 classes).

4.2. Preprocessing

Both datasets contain images that have unequal widths and heights, while our system requires square images of fixed size. We resize all images to a fixed resolution: the Leeds images to 96x96 pixels and the MPII images to 256x256 pixels. We also zero-center the data by subtracting the mean pixel value over the training set from each pixel over all three RGB channels. We do not normalize the data, since image data already has the same dimension size across all three dimensions: (0, 255). We augment the datasets by taking random crops and performing random horizontal flips of the training images. After data augmentation, the Leeds dataset consists of about 16,500 training and 1,000 validation images, and the MPII dataset consists of 180,000 training and 30,000 validation images.

4.3. Evaluation Metrics

For pose estimation, we evaluate using two different metrics, in order to compare our results with published results. The first metric is Percentage of Correct Parts (PCP) [2]. This metric measures the detection rate of limbs. A limb is considered correctly detected if the distance between the two predicted joint locations and the true limb locations is at most half the limb length. However, PCP penalizes shorter limbs, such as lower arms, which are harder to detect.

The second metric is Percentage of Detected Joints (PDJ) [10]. This metric considers a joint correctly detected if the distance between the predicted and ground truth joint locations is within a certain fraction (which we define and vary) of the torso diameter. We define torso diameter as being the distance between left shoulder and right hip. Using the PDJ metric means that all joint accuracies are evaluated using the same error threshold.

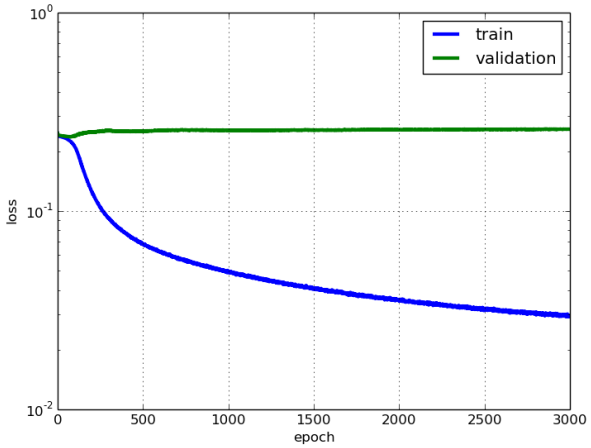

Figure 9. Training and validation loss for pose estimation plotted over 3000 epochs

PCP Training and Validation Accuracies						
	Head	Torso	Arms		Legs	
			Upper	Lower	Upper	Lower
Train	0.6637	0.6955	0.6846	0.5923	0.7515	0.7013
Validation	0.245	0.4982	0.2885	0.2945	0.5095	0.3085

Figure 10. PCP training and validation accuracies for different body parts

5. Results

5.1. Pose Estimation

After training for 3,000 epochs with the Mean Squared Error loss function, we ended up with 0.0577 training MSE and 0.104725 validation MSE (see Figure 9). Training loss decreases exponentially over all 3,000 epochs, but validation loss quickly bottoms out. This suggests that we are learning the training dataset well, but generalization plateaus. To address this, we could do more data augmentation in order to increase our dataset size, and/or construct a deeper CNN with more parameters.

We calculate the PCP on the Leeds Sports Pose Dataset for the head, torso, arms and legs (see Figure 10). We compare results for the most challenging limbs–upper and lower arms and legs–as well as the average across all challenging limbs, to five state-of-the-art approaches, as seen in Figure 11.

We also calculate the PDJ on the Leeds Sports Pose Dataset across the head, torso, arms, and legs, as seen in

PCP at 0.5 threshold on Leeds Sports Pose Dataset					
	Arms		Legs		Average
	Upper	Lower	Upper	Lower	
Ours	0.2885	0.2945	0.5095	0.3085	0.35
Dantone, et. al	0.45	0.25	0.65	0.61	0.49
Tian, et. al	0.52	0.33	0.70	0.60	0.56
Johnson, et. al	0.54	**0.38**	0.75	0.66	**0.58**
Wang, et. al	**0.565**	0.37	**0.76**	**0.68**	0.56
Pischulin	0.49	0.32	0.74	0.70	0.56

Figure 11. Percentage of Correct Parts (PCP) at 0.5 on Leeds Sports Pose Dataset for our model as well five state-of-the-art methods. These methods are from [1], [9], [4], [11], and [7], respectively.

PDJ Training and Validation Accuracies				
	Head	**Torso**	**Arms**	**Legs**
Train	0.8043	0.8111	0.7957	0.7790
Validation	0.6190	0.6338	0.4272	0.4564

Figure 12. PDJ training and validation accuracies for different body parts on threshold 0.5

12. Using the PDJ metric allows us to vary the threshold for the distance between ground truth joints and predicted joints.

Qualitative analysis. To get a better idea of the holistic performance of our algorithm, we visualize a sample of predicted poses on the test set, as seen in Figure 15. We can see that our algorithm is able to generate the correct pose for a variety of conditions: rotation (row 3, column 1 and row 4, column 1), figures that are turned sideways (row 2, column 1), foreshortening (the arms of row 5, column 3), occluded limbs (row 1, column 2 and row 2, column 5), and different lighting conditions (row 3, column 2 and row 2, column 4). Even when the prediction is not precise, our model usually gets the overall shape of the pose correct. Common errors include a failure to extend the arms and legs to their full lengths and confusing a person's orientation (i.e., whether or not they are facing the camera).

5.2. Activity Classification

We test our classification CNN on a validation set of approximately 3,000 distinct images (augmented to give 30,000 images). Classification of this validation set of into the 410 activity categories achieves a maximum accuracy of 31.89%. Classification by the 20 general activity categories achieves an accuracy of 80.51%.

The validation loss decreases quickly initially and appeares to plateau around 0.8 after 1,000 iterations, while the training loss continues to decrease until about the 2,000th iteration (Figure 16). Similarly, the classification accuracy

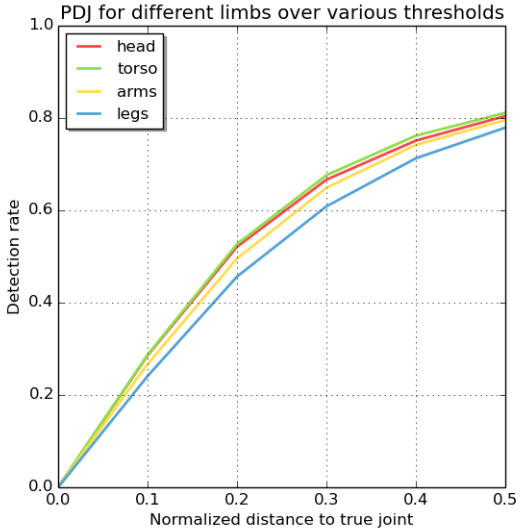

Figure 13. Training PDJ results presented over a range of normalized distances between predictions and ground truth labels. We plot over the range [0, 0.5] of the torso diameter. Results are accumulated into four categories: head (head and neck keypoints), torso (neck, shoulders, hips), arms (wrist, elbow, shoulders), and legs (ankles, knees, hips).

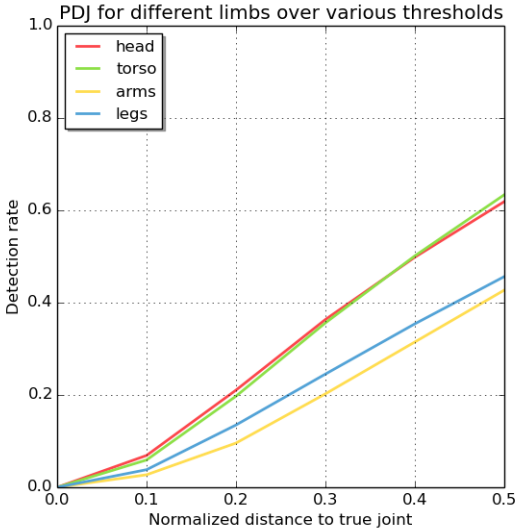

Figure 14. Validation PDJ results presented over a range of normalized distances between predictions and ground truth labels. We plot over the range [0, 0.5] of the torso diameter. Results are accumulated into four categories: head (head and neck keypoints), torso (neck, shoulders, hips), arms (wrist, elbow, shoulders), and legs (ankles, knees, hips).

(using the 20 general action categories) increases steeply and plateaued around 1,500 iterations, while the training accuracy continues to increase until about the 2,000th it-

Figure 15. Visualized predicted human pose trees on images from the Leeds Sports Pose Dataset. Each pose is represented as a stick figure where the colored lines connect predicted body keypoints. Different limbs are colored differently in the same image; each limb is colored the same across multiple images.

eration, as shown in Figure 17).

The relatively large gap between the training loss and validation loss may suggest that the model is over-fitting the training data or the loss function optimization is converging to a local minimum. We can address these issues by further fine-tuning the hyperparameters of our model (learning rate, step size, regularization strength, momentum), reducing the complexity of the model, and increasing the training set size through data augmentation.

Figure 18 shows the confusion matrix of the activity category classification results. Note that more distinctive and active activities, such as dancing, sports, music playing, and winter activities, have high classification accuracies, while broader categories, such as inactivity and volunteer activities, have lower classification accuracies.

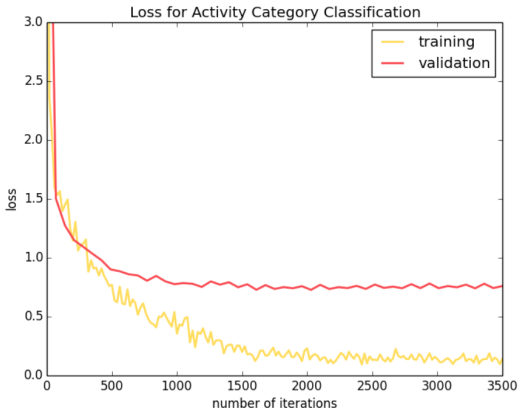

Figure 16. Activity classification training and validation loss on the 20 general action categories.

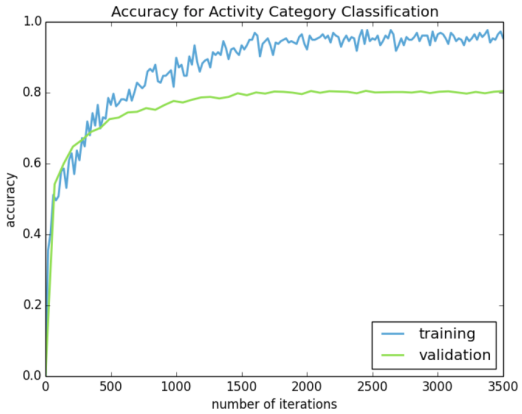

Figure 17. Activity classification training and validation accuracy on the 20 general action categories.

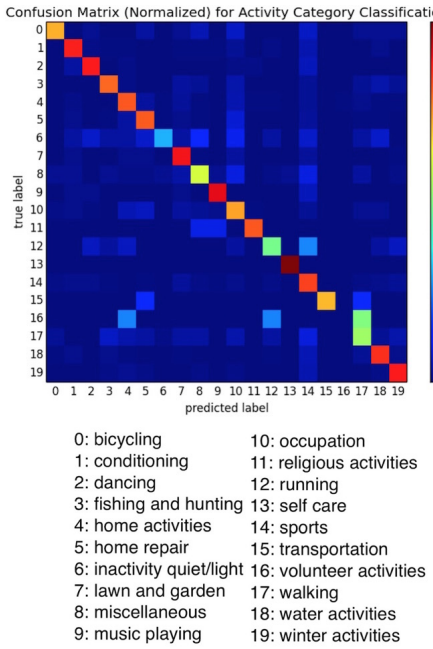

Figure 18. Confusion matrix, normalized by row, for activity classification on the 20 general action categories

6. Conclusion

Convolutional neural networks are a favorite architecture for computer vision tasks, due to their simplicity and intuitive nature, and their reduced number of parameters when compared to fully-connected models. They have been applied, with great success, to image classification tasks. It is only recently that computer vision researchers have begun applying CNNs to regression tasks. CNNs have the advantage of being a holistic model that takes the entire image as an input signal for each body keypoint, in contrast to local detectors, whose reasoning is constrained to a single part and can only model a small subset of interactions between body parts.

We showed that human pose estimation can be cast a regression problem and modelled with a generic CNN. Our application of CNNs to the problem of human pose estimation achieves competitive results on a challenging academic dataset with a simple model. We hypothesize that we would be able to achieve even better results with more compute power and space (the depth of our regression CNN was limited by the RAM of the GPU it was trained on).

6.1. Future Work

To decrease the gap between training and validation performance further fine-tune the hyperparameters of our model. Things to consider include adjusting the base learning rate and learning rate policy, tyring different types of mementum updates, and tuning the regularization strength. Furthermore, model ensembles can be used to increase performance.

We would also like to experiment with a combination of joint estimation and activity classification tasks to see if knowing the locations of joints in an image improves the activity classification performance of a CNN. We would first run the input image through the joint estimation regression model to obtain the human pose information, and use this as a secondary input (in addition to the original input image) into the classification model. This additional information may help our model determine the activity being performed in the image. However, it is possible that 2D pixel coordinates will not provide sufficiently useful information regarding the true 3D pose and action type.

Acknowledgments

We would like to acknowledge the CS231N instructors, Fei-Fei Li and Andrej Karparthy, and staff for their guidance with this project. We would also like to thank Keenon Werling for the use of his GPU.

References

[1] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. pages 3041–3048, June 2013.

[2] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214, 2012.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[4] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1465–1472, June 2011.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[6] D. Nouri. Using convolutional neural nets to detect facial keypoints.

[7] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 588–595, June 2013.

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[9] Y. Tian, C. Zitnick, and S. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 256–269. Springer Berlin Heidelberg, 2012.

[10] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. pages 1653–1660, 2014.

[11] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 596–603, June 2013.