# Human Pose Estimation Using Convolutional Neural Networks

Amy Bearman
Stanford University
abearman@cs.stanford.edu

Catherine Dong
Stanford University
cdong@cs.stanford.edu

## Abstract

## 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. Human pose estimation has made significant progress in the last few years, especially with the introduction of Deep Convolutional Neural Networks (CNN). The identification of body keypoints has proved to be a challenging problem due to small joints, occlusions, and the need to capture context [2]. In this project, we explore different CNN architectures for modeling human pose estimation.

### 1.1. Related Work

At the high level human activities can often be accurately characterized in terms of body pose, motion, and interaction with scene objects [Pishchulin1]. 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. [2] that applying deep convolutional neural networks (CNN) to pose estimation as a regression problem has the advantage of reasoning about pose in a simple but hollistic 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 higher PCP score (0.61) and was much simpler than previous methods based on explicitly designed feature representations and graphical models.

### 1.2. Convolutional Neural Networks for Regression

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. [1].

## 2. Problem Statement

The data for this project comes from the MPII Human Pose Dataset, which contains about 25,000 images with over 40,000 people with annotated body joints (two x-y coordinates for the head rectangle, and x-y coordinates for each joint) and activity types. Each image is extracted from a YouTube video. There are images from across 410 different human activities, and each image is provided with an activity label.

Our problem consists of two tasks: Body keypoint identification and action classification. For body keypoint identification, our network will take as input the raw MPII image and output a vector of coordinates of the body keypoints. We will try three methods of accomplishing these tasks: First, since these coordinates are real-valued numbers, we will formulate this part as a regression problem. The action classification network will then use these keypoint coordinates as well as the original input image to determine the type of action being performed in the image. Second, we will try the reverse – we will attempt to classify the image and use the predicted label to assist in keypoint identification. Third, we will try combining the tasks of classification and keypoint estimation using a CNN with an output layer that represents keypoint estimates for each class.

## 3. Technical Approach

### 3.1. Data

We are using the MPII Human Pose Dataset, which contains about 25K 1280x720 pixel images with over 40K people with annotated body keypoints. Body keypoints consist of two $(x, y)$ coordinates representing the head rectangle, and $(x, y)$ coordinates for each joint. Each image is extracted from a YouTube video. There are images from across 410 different human activities, and each image is provided with an activity label. There are 20 broad activity types that we attempt to classify (such as ”sports,” ”transportation,” and ”music playing”).

## 3.2. Preprocessing

The MPII Human Pose Datset contains images that have unequal widths and heights, while our system requires smaller images of constant size and equal widths and heights. So, we down-sample the images to get a smaller, fixed resolution of 256x256. 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).

## 3.3. Model

### 3.3.1 Activity Classification

We formulate the activity classification problem as a multi-class classification problem that can be modelled by a CNN. The CNN takes as input a full image (256x256 pixels) and outputs the activity classification (indexed 0 through 19, representing 20 different human activities).

### 3.3.2 Pose Estimation/Joint Localization

We formulate the human pose estimation problem as a regression problem that can be modelled with a generic convolutional neural network. The CNN takes as input a full image (256x256 pixels) and outputs the pixel coordinates of each body keypoint. We are using the Caffe framework to implement our CNN. 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 **y** is the ground truth pose vector.

Instead of using a classification loss function (i.e., softmax or SVM), we'll use the L2 distance between our pose vector and the ground truth pose vector:

$$L = \sum_{(x,\mathbf{y}) \in D} \sum_{i=1}^{k} \|\mathbf{y}_i - f(w_i; x)\|_2^2 \tag{1}$$

where $D$ is the training set, $k$ is the number of body keypoints, and $w_i$ is the weights learned for the $i$-th body keypoint. We use backpropagation to optimize for the weights $w$.

### 3.3.3 Architecture

We plan to experiment with many different architectures, but the general form of the CNN layers should be as follows:

1. INPUT
2. (CONV, RELU)*N
3. POOL
4. (FC, RELU)*K
5. FC

Furthermore, we will use cross-validation to tune the hyperparameters which include

1. Learning rate
2. Batch size
3. Regularization
4. DropOut
5. Filter size

## 3.4. Other things we plan to try

- Calculating the bounding box for each human figure and normalizing all joints with respect to this box
- Adding random perturbations of the input images to generate more data
- Experimenting with different sizes and numbers of filters
- Adding DropOut
- Trying other activation functions such as leaky ReLU or maxout
- Using model ensembles
- Alternative update steps (i.e., momentum, Nesterov momentum, RMSprop, AdaGrad, AdaDelta)
- Defining our own evaluation metric (other than PCP or PDJ)

## 3.5. Evaluation

Obviously, we cannot only consider a joint prediction correct if it exactly matches the ground truth label. We will consider two different metrics for pose estimation performance. First is the Percentage of Correct Parts (PCP) metric, which measures the detection rate of limbs, where a limb is considered to be correctly detected if the distance between the two predicted joint locations and the ground truth locations is at most half of the limb length. However, the PCP metric penalizes shorter or foreshortened limbs, which is a drawback.

To address this drawback, we will also evalute using the Percent of Detected Joints (PDJ) metric, which considers a joint correctly detected if the distance between the predicted and ground truth joint is within a certain fraction (which we will define and vary) of the torso diameter. PDJ means that all joint accuracies are evaluated using the saem error threshold.

## References

[1] 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.

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