CS230

Human Pose Estimation using Convolutional Neural Networks

Richard Hsieh
Department of Mechanical Engineering
Stanford University
rhsieh91@stanford.edu

Abstract

The task of human pose estimation has interesting applications for activity classification and body movement predictions. Inspired by previous work using Convolutional Neural Networks (CNN), this paper explores an end-to-end deep learning model that can take an input image and output positions of key joints on a human figure. A 11-layer CNN is built that has shows promising trends but could still use further improvement with bias and especially variance. Sample predictions are inspected and saliency maps are analyzed to provide intuition on what the model is lacking. These error analysis techniques lend confidence that the model is trending in the right direction and provide further insight into next steps for improvement.

1 Introduction

Human pose estimation involves identifying key joints and limbs on the human body, which can be useful for activity classification and body movement predictions. Traditionally, this has been a challenging problem due to joints being small/obscured and the need for contextual understanding of the human body in question. Convolutional Neural Networks (CNN) provide an end-to-end learning approach for tackling this problem without the need for hand-crafted features. In this case, the input is a RGB image containing a human figure(s) and the outputs are (x,y) coordinate pairs for key joints on the figure (e.g. elbows, shoulders, hips, etc.). By drawing line segments between key joints, one can then superimpose limbs and body parts onto the human figure.

2 Related work

There is significant interest in human pose estimation in the research community. In the 2D static image domain, Tosehev et. al. formulated human pose estimation and joint localization as a regression task and approached it using a deep convolutional neural network (CNN) [1]. Previous students in CS231n: Convolutional Neural Networks for Visual Recognition have also worked on this exact problem statement too. Bearman and Dong trained two independent deep CNNs to tackle pose estimation as a regression problem and activity classification as a multi-class classification problem [2].

3 Dataset & Preprocessing

The dataset used is the Leeds Sports Pose Dataset [3] and the Leeds Sports Pose Extended Dataset. Together, this contains 12,000 images (as shown in Fig. 1) cropped and scaled to focus on prominent

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

human figure engaging in a sport or activity. Each image is annotated with the *(x,y)* coordinates of 14 key joints: right/left ankle, right/left knee, right/left hip, right/left wrist, right/left elbow, right/left shoulder, neck, and head top. The dataset set is further augmented to a grand total of 24,000 images by flipping each image about its vertical axis.

Figure 1: Sample Images in Leeds Sports Pose Dataset

The dataset is divided using a *90-10* train/test split, which provides a sufficient test set of 2400 images. Since the images are of irregular width and height, each image is also rescaled to 96×96 pixels (without maintaining aspect ratio) prior to training. Lastly, no pixel normalization is performed since the pixel intensities are already fixed within a range of $(0, 255)$ across the red-green-blue (RGB) channels.

4 Methods

4.1 Models

A couple different CNN models were explored using different combinations of convolutional, max pooling, and fully connected layers with and without batch normalization. The most promising model is illustrated in Fig. 2. It consists of convolutional layers with filter sizes of 3×3 and a stride of 1. The number of filters is increased from 32 to 64 to 128 over the 3 convolutional layers. Prior to the ReLU activation in each convolutional layer, a batch normalization layer is also added. The max pooling layers have a filter size of 2×2 with a stride of 2. Lastly, the fully connected layers each have 500 neurons with ReLU activations except for the final layer which is linear. Similar to the convolutional layers, batch normalization layers are added prior to each ReLU activation.

Figure 2: 11-layer CNN with Batch Normalization and Dropout

4.2 Training Details

Since the model aims to predict exact *(x,y)* coordinates, the problem is formulated as regression task with a mean squared error loss function. The models weights are initialized with Xavier initialization at the beginning of training. Dropout is used (with a rate of 0.5) after each max pooling and fully connected layer. The learning rate is set adaptively using the Adam optimization algorithm training on mini-batches of 128.

5 Results & Discussion

5.1 Training Loss

The model was quite difficult to train and unfortunately the final results still indicate a high bias and variance. On the bright side, the inclusion of batch normalization had a significant effect in reducing training loss and improving training accuracy as shown in Fig. 3. However, the test loss and accuracy plateau quickly after only a few epochs while training loss continues trending downwards and training accuracy trends upwards. This indicates that the model is in fact able to overfit the training data and that the more pressing issue is with improving variance.

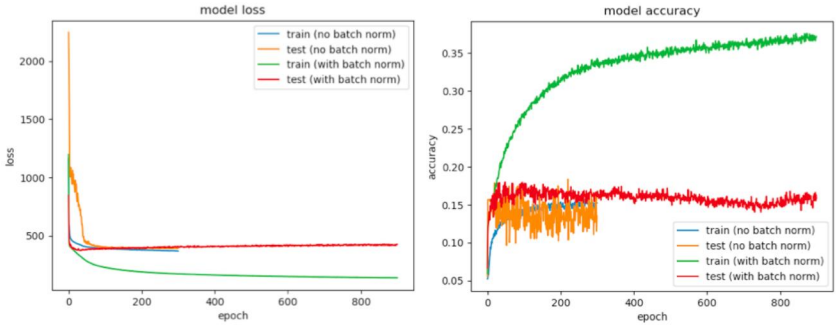

Figure 3: Loss and Accuracy Comparisons with and without Batch Normalization

5.2 Sample Predictions

By visually inspecting some of the predictions made by the model (Fig. 4, one can qualitatively identify key trends and downfalls in the model.

Figure 4: Sample Predictions from Test Set

As shown by the leftmost figure, when there is a single prominent human body, the model is able to correctly estimate a human figure and not just a random shape. The model also appears to mimic the general pose of the body as evidenced by the left arm of the human subject and estimated pose being further away from the torso than the right arm.

On the other hand, when a image has multiple human bodies, the model has difficulty locking onto the main subject as seen in the center figure. Additionally when the human figure is contorted as in the rightmost figure, the model has great difficulty distinguishing and separating the key joints from each other.

5.3 Saliency Maps

A useful tool in understanding what the CNN is identifying at different layers is through saliency maps. Saliency maps compute the gradient of the layer of interest with respect to the inputs. In

essence, this highlights which pixels in the images are most activated and provides intuition for what a layer is trying to identify. In this case, the saliency map of the 3rd convolutional layer in Fig. 2 is of interest.

As shown in Fig. 5, the 3rd convolutional layer appears to roughly activate on (i.e. identifying) edges in the picture. It appears to identify the edges for the arms of the main human subject but also identifies the edges in the yellow post by the human's legs. This explains why in the leftmost image in Fig. 4, the model seems to learnt the general pose of the arms but not the legs.

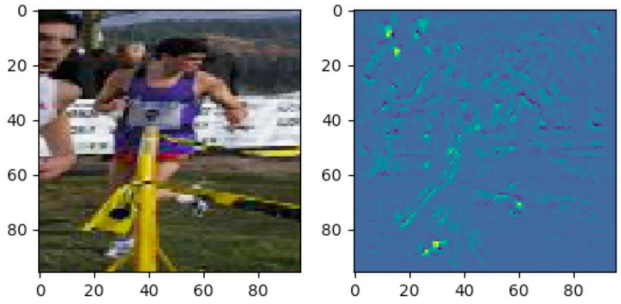

Figure 5: Saliency Map for Image in Test Set - 1

Furthermore, in Fig. 6 one can also see that the 3rd convolutional layer is correctly identifying the edges of the human figures. More interestingly, this layer seems to activate strongly on the wrists of the main human subject (person jumping in the air) but also on the heads of the humans in the distance. This indicates that the model can not discern properly between smaller, finer features. As such, this explains why in the center image of Fig. 4 the model is locked onto the humans in the background as it may be incorrectly identifying the heads of these humans as different joints on a single human.

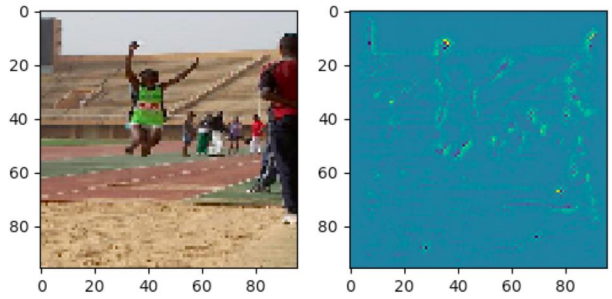

Figure 6: Saliency Map for Image in Test Set - 2

6 Conclusion & Future Work

There is certainly room for improvement for both the bias and variance of the model. This can be achieved by further tuning the architecture and depth of the CNN as well as experimenting with various regularization techniques (e.g. L1 or L2 regularization, varying dropout rates, etc.). However, from analyzing the saliency maps, one can see that the current model is in fact learning something useful to the task at hand. This lends confidence to the fact that a deeper and more complex architecture compared with a longer training time could significantly improve the bias of the model.

One interesting extension of this pose estimation model would be in activity classification. Intuitively, it seems that pose information would be highly useful in classification tasts involing human activities. With a robust pose estimation model, one can take an intermediate layer and merge with another

intermediate layer of pre-trained image classifier to see if classification accuracy can be greatly improved.

7 Contributions

Richard Hsieh worked on this project and report with the mentorship of Shervine Amidi.

References

[1] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[2] Amy Bearman and Catherine Dong. Human pose estimation and activity classification using convolutional neural networks. *CS231n: Convolutional Neural Networks for Visual Recognition (Final Project Report)*.

[3] Sam Johnson and Mark Everingham. 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.