CS 148: Pose-Augmented Activity Recognition

Michelle Zhao[1]
California Institute of Technology[1]
mzhao@caltech.edu

Abstract

This project proposes a novel training framework for the activity recognition problem based on a pose estimation intermediate step, and convolutional neural networks (CNNs). The pose estimation problem is formulated as a CNN-based regression problem towards body joints. I use a CNN regressor to generate high precision pose estimates of images before performing activity recognition. I add a mask over the image of joint location estimates as a fourth layer onto the RGB image. Using image + mask, this approach has the advantage of reasoning about the image subject's pose before attempting to classify the activity that the subject is performing. This algorithm, called PoseMask, benefits from the intermediate mask-generation step by forcing the model to interpret the subject's pose, and discourages the model to classify based on potentially unrelated aspects. The PoseMask framework has is a simple, yet powerful formulation which capitalizes on recent advances in pose estimation for the classification problem. This project presents a proof-of-concept of this framework on diverse real-world images.

1. Introduction

The objective of human pose estimation is to locate the joints, or key-points, of one or more figures in an image. Some examples of key-points of the human pose include the head, elbows, hips, hands, and feet. This problem of joint localization becomes increasingly difficult when the subject in the image is contorted in different poses, when the some body parts are occluded in the image, and when operating on rich, dense images. The human pose orientation is represented in a graphical format, specifically a set of coordinates that can be connected to describe the pose of the person. Each co-ordinate in the skeleton is a part or joint. Limbs of the person are formed by connecting valid pairings of key-points.

There have been several data sets designed for the pose estimation problem, including but not limited to the MPII Human Pose Dataset and the Leeds Sports Pose Dataset. In this work, I utilized the Leeds Sports Pose Extended Training Dataset, a dataset of 10,000 images annotated with up to 14 visible joint locations.

Figure 1. Example of human pose estimation objective.

The task of 2D articulated human pose estimation in still images has been widely explored. One class of pose estimation methods uses graphical models which represent the human body as a connected collection of parts corresponding to head, torso and other limbs. [2] models the probability of a part being present at a particular location and orientation given the input image. Convolutional Neural Networks (CNNs) have performed well on the pose estimation problem. Toshev et al. [1] demonstrated that pose estimation as a joint regression problem can be successfully cast in DNN settings. This DNN formulation benefits from the advantage of being able to capture the full context of each body joint, since each joint regressor uses the full image as a signal. The success of DNNs suggests an ability to naturally perform some form of holistic reasoning over the entire subject body that enables prediction of occluded joint locations. Furthermore, this approach is more simple than other graphical methods, because there is no need to design feature representations of the image or part-space.

The problem of activity recognition is a classification problem that aims to identify a pre-defined set of physical actions from images or videos. Activity recognition from images, the focus of this work, is a difficult problem because human motion is captured through a sequence of video frames, but this motion sequence is lost when working with image-only classification. Humans are generally able to identify activity from images using information priors and latent image understanding. This is a complex decision-making process for neural networks to establish. Thus, this work aimed to leverage understanding of the subject's pose to better interpret activity from images.

2. Problem Statement

Pose estimation datasets aim to identify a list of body part locations in the image as (x, y) image coordinates. I used the Leeds Sports Pose Extended Training Dataset, which is annotated with 14 joint locations: right ankle, right knee, right hip, left hip, left knee, left ankle, right wrist, right elbow, right shoulder, left shoulder, left elbow, left wrist, neck, and top-of-head. See Figure 2 for an example.

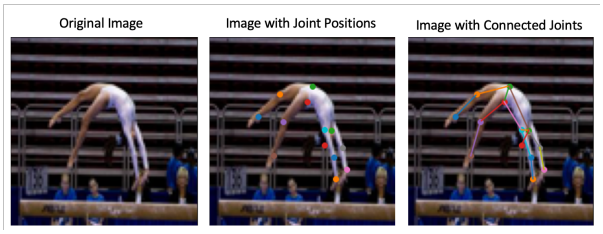

Figure 2. Leeds Training Example

The 14 annotated training joints $\{(x_1, y_1), ..., (x_{14}, y_{14})\}$ are flattened into a single vector $(x_1, y_1, x_2, y_2, x_{14}, y_{14})$, which serves as the desired output. The objective of the pose estimation problem is to identify xy-coordinates for all 14 body joints of the subject in the image. For the activity recognition problem, I train a baseline network that takes as input an image, and outputs a softmax probability distribution over 8 possible activity classes: rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock-climbing.

Figure 3. Activity Recognition Outputs: UIUC Sports Dataset

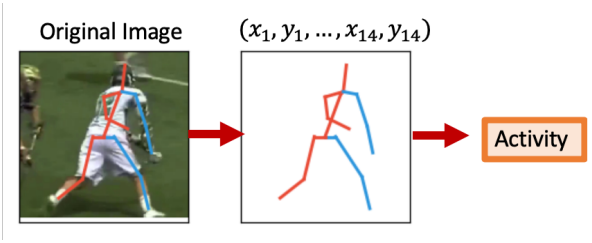

Figure 4. Activity Recognition Outputs: UIUC Sports Dataset

The final objective is to augment the input of the same activity classification network with pose estimates, and classify the activity (Fig 4). My minimum goal was to trained a human pose regressor on images. My medium goal was to train a baseline activity recognition network that takes as input only images. My final goal was to train an activity recognition network that takes as input both images and pose.

3. Related Work

Human action recognition from images is an active research area in computer vision and pattern recognition. The problem objective is to identify a person's action or behavior from a corpus of possible action classes. Image-only action recognition poses an interesting challenge in that there are no available spatio-temporal features, as there are with video action classification. Thus, a classifier must either infer temporal features of the image or make classifications based solely on spatial elements of the single-frame. Furthermore, many naturally existing images are cluttered and complex, so robust and efficient action recognition methods for still images remain in development.

The four main classes of approaches to solving action classification are model-based approaches, example-based methods, pictorial structure-based methods and poselet based approaches [5]. The model-based approach uses a known parametric body model is used for matching pose variables[4], and applies spatial and functional constraints on each of the perceptual elements. Example-based models, like [6], use clustering or classical machine learning algorithms to group images into action categories. My work is in the vein of pictorial structure-based methods, in which pose representation is treated as a feature for action classification. [7] is a similar method to this work, which uses pose estimation as a feature for action classification, but their method uses decision trees instead of convolutional neural networks. My work is also similar to poselet-based methods, which use annotated three-dimensional images for training [8]. The difference is my annotations are a pose-mask.

Algorithm 1: PoseMask: Pose-Augmented Action Recognition

ϵ = training error bound;
Dataset $D = \{(F_i, (x_1^i, y_1^i, ..., x_{14}^i, y_{14}^i))\} \forall i$;
Generate poses $P_i \leftarrow$ PoseEstimator(D_i) $\forall i$;
Concatenate to new inputs $G_i = (F_i \cup P_i)$;
Initialize action recognition network M;
Split train/test;
while *Training Error* $> \epsilon$ **do**
    Train the model M using T ;
    Forward-propagate M on training set;
    Back-propagate M with Negative Log Loss;
end

4. Algorithm

The algorithm, Algorithm 1, first estimates the pose of an input image, then creates a mask over the image containing the subject's estimated pose.

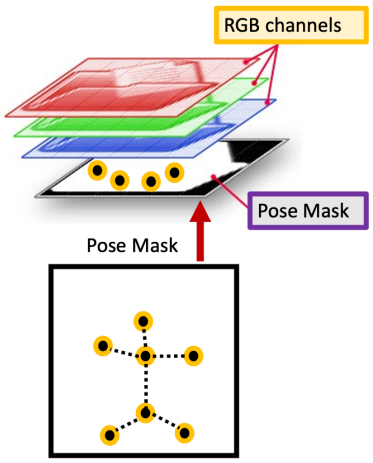

Figure 5. Concatenation to form 4-channel input.

The mask is concatenated with the RGB channels of the original image to create a 4-channel input to a convolutional neural network that outputs softmax action class predictions. The process is illustrated in Figure 6.

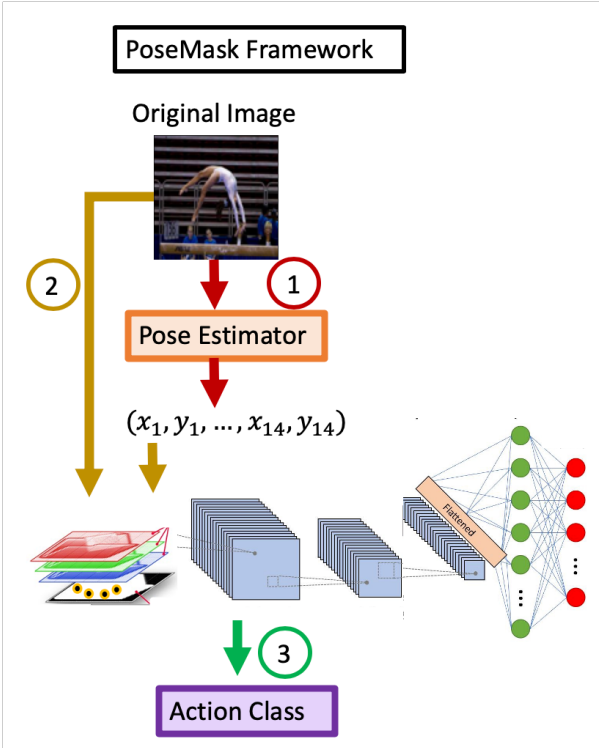

Figure 6. Diagram of PoseMask framework.

5. Approach

I ran three separate experiments to establish the pose-augmented action recognition framework, called PoseMask. First, I created a pose estimator. Second, I trained a baseline convolutional action recognition neural network. Third, I trained the same network as the baseline CNN, with the only difference being that the PoseMask network takes in a 4-channel input.

5.1. Pose Estimation

Model Architecture For pose estimation, I used the AlexNet architecture [9]. AlexNet is a CNN architecture, designed by Krizhevsky and Hinton. AlexNet achieved a top-5 error of 15.3% on the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012. I chose to use AlexNet due to its proven ability to understand visual features.

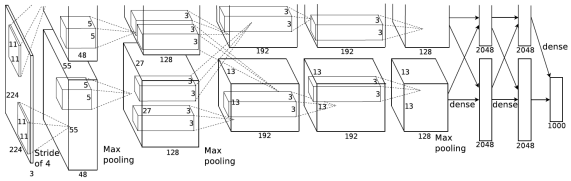

Figure 7. Diagram of AlexNet model architecture from [9]

The architecture consists of eight layers: five convolutional layers and three fully-connected layers. Relu is applied after very convolutional and fully connected layer. Dropout is applied before the first and the second fully connected year. AlexNet also introduces overlapping pooling, instead of traditional CNN pooling over neighboring groups of neurons with no overlapping.

Training Details In order to model pose estimation as a regression problem, I flatten the (14×2) joint coordinates

$$\{(x_1, y_1), ..., (x_{14}, y_{14})\}$$

into a single (28×1) vector

$$(x_1, y_1, x_2, y_2, x_{14}, y_{14}).$$

Due to the regression objective, MSE loss between predicted and target joint coordinated was used.

$$P_i = (x_1^i, y_1^i, x_2^i, y_2^i, x_{14}^i, y_{14}^i)$$

$$\hat{P}_i = f(H) = (\hat{x}_1^i, \hat{y}_1^i, \hat{x}_2^i, \hat{y}_2^i, \hat{x}_{14}^i, \hat{y}_{14}^i)$$

$$MSE(f) = \frac{1}{N}\sum_{i=1}^{N}(P_i - \hat{P}_i)^2$$

$$MSE(f) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2$$

where $K = 14$, H_i is the input image, and N is the number of training examples.

The AlexNet model was trained with a batch size of 256 and learning rate of 0.01 for 5000 epochs. I used backpropagation to optimize for the model weights. Specifically, Adagrad was used for the optimizer. Adagrad is an algorithm for gradient-based optimization that adapts the learning rate to the parameters, performing smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequent features. For this reason, it is well-suited for dealing with sparse data.

5.2. Baseline: Activity Recognition

Model Architecture The baseline action network is a simple convolutional neural network with 3 convolutional layers and 3 fully connected layers. Relu activation and max pooling is performed on every convolutional layer. Layer normalization is performed on only the first two convolutional layers.

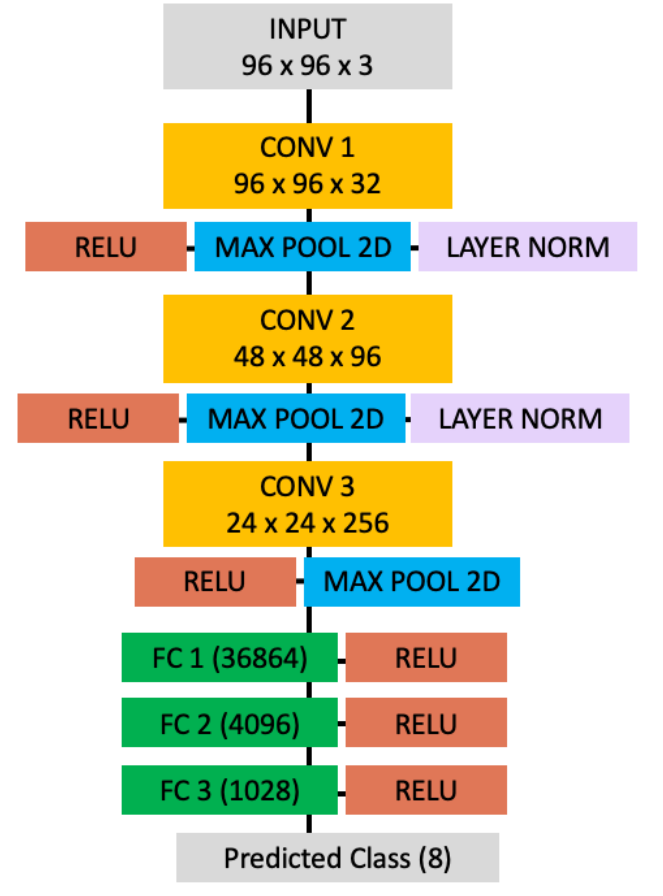

Figure 8. Diagram of baseline action network model architecture.

Training Details The 8 action classes are: rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock-climbing. The model was trained using Negative Log Loss (NLL) for backpropagation. NLL is effective for multi-class classification problems.

$$NLL(f) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})]$$

where N is the number of training examples.

The baseline action recognition model was trained with a batch size of 256, learning rate of 0.01, and momentum of 0.9 for 100 epochs. I used backpropagation and stochastic gradient descent to optimize for the model weights. I found that the model didn't need all 100 epochs, and was completed training in about 35 epochs.

5.3. Pose-Augmented Activity Recognition

Model Architecture To prove the efficacy of the PoseMask framework, I wanted to trained the same network as the baseline CNN, with the only difference being that the PoseMask network takes in a 4-channel input. This way, I keep every variable of the experiment controlled, and the only variable changed is the experimental variable: the augmented 4-D input with the pose-mask. If I change the network as well, then differences in training rate and test error are potentially due to having a more complex network rather than due to the pose estimation intermediate.

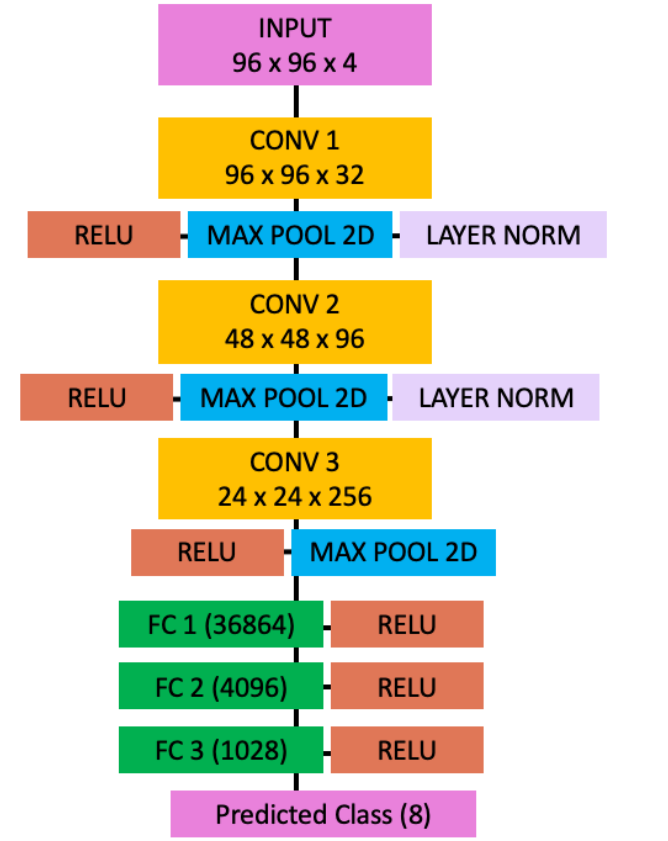

Figure 9. Diagram of PoseMask augmented action network model architecture.

Training Details This model is trained similarly to the baseline action recognition model. I use Negative Loss Loss

and train with a batch size of 256, learning rate of 0.01, and momentum of 0.9 for 100 epochs. I used backpropagation and stochastic gradient descent to optimize for the model weights.

6. Setup

6.1. Data

For action recognition, I used the UIUC Sports Event dataset, which was developed in conjunction with the Stanford Vision Lab. The dataset contains 8 sports event categories: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). The images were divided into a 95%-5% train-test split.

For pose estimation, I used the Leeds Sports Pose Dataset, which contains 10,000 pose annotated images of mostly sports people gathered from Flickr using the tags shown above. The images have been scaled such that the most prominent person is roughly 150 pixels in length. Each image has been annotated with 14 joint locations. Left and right joints are consistently labelled from a person-centric viewpoint. The images were divided into a 95%-5% train-test split.

6.2. Pre-processing

Because the images for both datasets were non-uniform, I resized all of the images to 96×96. I then zero-center all data over all three RGB channels. The Leeds dataset consisted of about 9,500 training and 500 test images, and the UIUC action dataset consisted of 1,500 training and 100 test images.

6.3. Evaluation Metrics

For pose estimation, I evaluate two metrics. First, I use the mean-squared-error between the 14 target and estimated joint locations. The second metric is Percentage of Correct Keypoints (PCK). This metric considers a keypoint correctly detected if the distance between the predicted and ground truth joint locations is within a certain threshold, defined to be the distance of the torso diameter.

For action recognition, I measure overall classification accuracy.

$$Acc = \frac{1}{N} \sum_{i=1}^{N} 1_{y_i = f(h_i)}$$

I also compare performance of the baseline and PoseNet frameworks using their confusion matrices.

7. Results

7.1. Pose Estimation

The pose estimation model was trained over 5000 epochs. The AlexNet structure worked decently for pose estimation. However, had there been more time, more epochs of training would improve the error.

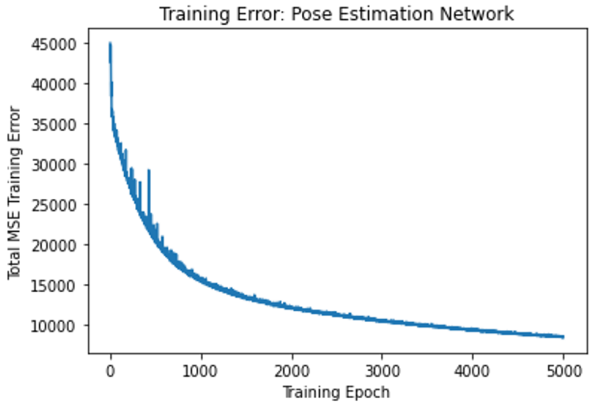

Figure 10. Training error of pose network.

The pose estimation network predicted too heavily the forward-facing anatomical human pose, which is likely because of the prone human pose being the dominant distribution of body parts in the dataset.

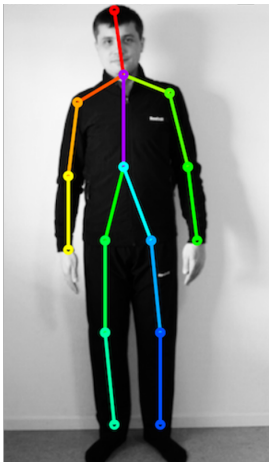

Figure 11. Example of forward-facing anatomical pose.

Below are two examples of Pose Estimation Network output predictions. The predicted poses are still inaccurate, and the model heavily favors predicting the forward-facing anatomical pose.

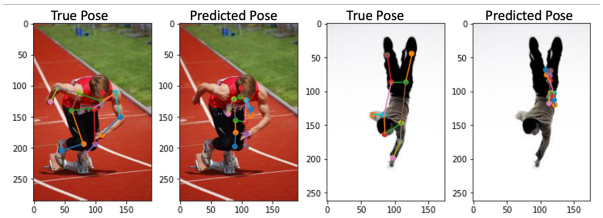

Figure 12. Example of pose network predictions.

Taking a closer look at the upside down predicted pose, I can see that the pose network was able to recognize that the human subject was inverted, and the pose it predicts is an upside-down figure. This is a fairly good result, because I can see that the pose network is able to interpret the image and human pose.

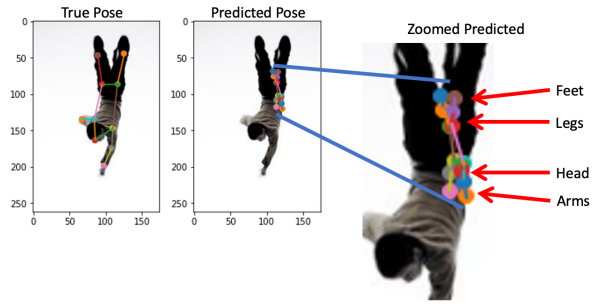

Figure 13. A closer look at the upside down predicted pose. This is a good result.

The mean-squared-error test loss is 2401.75. The MSE train loss is 6917.43. The PCK of the pose estimator shows that the pose estimator reaches 80% correct keypoints when the allowed threshold is relaxed to 50 pixels, which is poor performance.

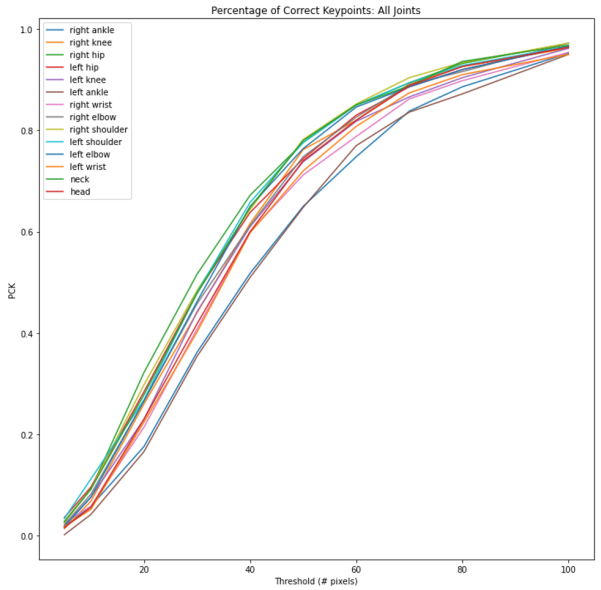

Figure 14. Percentage of Correct Keypoints.

7.2. Activity Recognition

I compared two activity recognition baselines: a 3-convolutional-layer baseline neural network and a 4-convolutional-layer baseline network. The 3-Conv layer baseline performed significantly better than the 4-Conv layer baseline. This is likely due to large size of the 4-layer network, which would cause it to train more slowly.

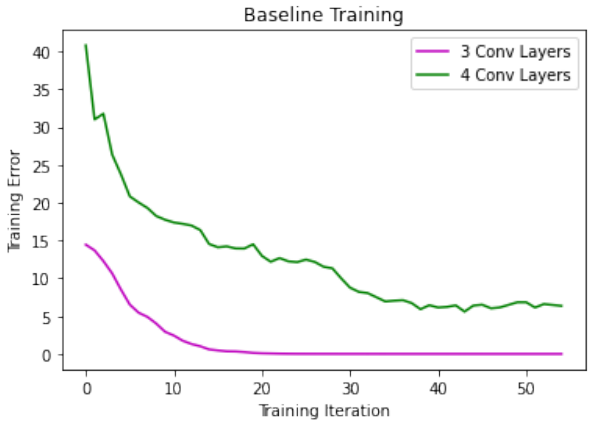

Figure 15. Training error of baseline action recognition network.

From this study, I chose to move forward with the 3-convolutional layer baseline for pose-augmented action recognition. To keep variables constant, I used the same network as the 3-convolutional layer baseline, with the only difference in the PoseMask framework being that the input is 4-channels instead of 3. The NLL test loss is 0.003. The NLL train loss is 0.003. The confusion matrix of the network shows near perfect performance. Test accuracy was 1.0.

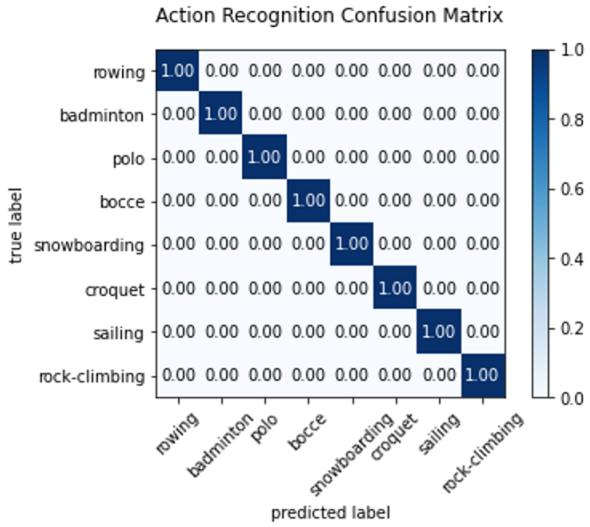

Figure 16. Confusion matrix of baseline action recognition network.

7.3. Pose-Augmented Activity Recognition

First, I constructed the pose-augmented data. Using the trained pose estimation network, I generate human pose etimates for each of the UIUC Sports Action dataset images. Using this now 4-channel input, I retrain the action recognition network, and compare the rate of training. If the rate of training increased, then this would indicate that the pose estimates are a beneficial intermediate for helping in identifying actions from images.

Below are some of the well-predicted pose estimate outputs on the UIUC action images. The model too heavily favors predicting the forward-facing anatomical pose, and cannot identify other contorted poses. I predict that this is likely due to dataset skew. The dataset likely has a high proportion of forward-facing prone poses, and fewer examples of other atypical poses. Although the poses are inaccurate, the network seems to function very well as a human-identifier. The network identifies the main human subject in each image very well.

Figure 17. These are examples of well-performing Pose Estimation Network output predictions. The predicted poses are still inaccurate, and the model heavily favors predicting the forward-facing prone pose.

Below are some of the most poorly-predicted outputs.

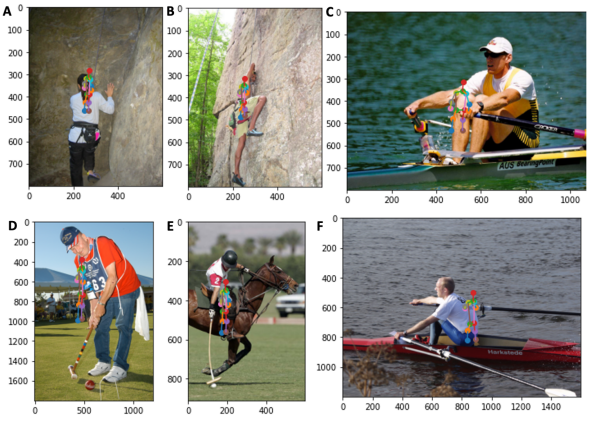

Figure 18. These are examples of poor-performing Pose Estimation Network output predictions.

The model sometimes fails to predict the pose or the scale of the human subject completely. The model is also unable to handle multiple subject inputs, because the Leeds data annotations from which the model was trained is only provided for the main subject. The NLL test loss is 0.026. The NLL train loss is 0.026. The PoseMask network had approximately the same performance as the baseline network. Its test accuracy was 0.999, with an identical confusion matrix as the baseline.

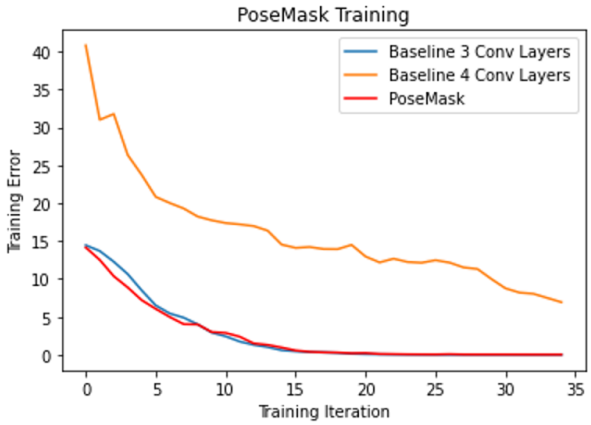

Figure 19. Training error of PoseMask vs. baseline action recognition networks.

The pose-augmented framework trained the action recognition network more quickly than the baselines. The PoseMask framework trained significantly faster than the 4-convolutional layer baseline. The PoseMask framework trained at about the rate as the 3-convolutional layer baseline. The training error decreased more quickly using the pose augmented input, but after about 10 training epochs, the baseline seemed to catch up.

Had the pose estimation network been more accurate, the pose-augmented action recognition network may have more significantly outperformed the baselines. I can conclude that the pose-augmentation only helps, and doesn't hurt the training of the action recognition network. This work also shows that a lightweight pose estimator is a viable intermediate for image understanding problems involving human subjects.

8. Future Work

Due to the poor performance of the pose estimation network, my next steps are testing action classification training with a pre-trained pose estimator, like DeepPose. I'd also like to try the PoseMask action recognition framework on a larger action images dataset. Eventually, this work could be extended to augmenting video understanding as well.

9. Implementation

My code can be found at `https://github.com/mzhao98/pose-estimation`. Note this is a different repo than my Homework 4 repo. I cleaned this one up for submission.

10. Conclusion

This work proposed the a CNN-based training framework for still-image action recognition that performs pose estimation intermediate step. First, a CNN regressor generates pose estimates of images, which forms a pose-mask over the image. The pose-mask forms the fourth layer stacked onto the RGB image. Using image + mask, this approach has the advantage of reasoning about the image subject's pose before attempting to classify the activity that the subject is performing. This framework, called PoseMask, forces the model to interpret the subject's pose, and discourages the model to classify based on potentially unrelated aspects. This work showed a proof-of-concept of the PoseMask framework on diverse real-world images, and encourages further work in the pose-augmented understanding of visual data.

11. Acknowledgements

I would like to thank Professor Pietro Perona and CS148 teaching assistants Elijah Cole and Serim Ryou for their guidance in research and teaching throughout the term.

12. Bibliography

1. A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1653-1660, doi: 10.1109/CVPR.2014.214.
2. S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In BMVC, 2010.
3. Li-Jia Li and Li Fei-Fei. What, where and who? Classifying event by scene and object recognition . IEEE Intern. Conf. in Computer Vision (ICCV). 2007.
4. A. Gupta, A. Kembhavi, L.S. Davis,(2009) Observing human-object interactions: using spatial and functional compatibility for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) 17751789.
5. Guo, Guodong and Alice Lai. "A survey on still image based human action recognition." Pattern Recognit. 47 (2014): 3343-3361.
6. Y. Wang, H. Jiang, M.S. Drew, Z.N. Li, G. Mori, (2006), Unsupervised discovery of action classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR2006, pp. 16541661
7. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, (2011), Real-time human pose recognition in parts from single depth images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR2011, pp. 12971304
8. G. Sharma, F. Jurie, C. Schmid,(2013), Expanded parts model for human attribute and action recognition in still images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR2013, pp. 18.
9. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.