Translating 2D Pose Heatmaps to Continuous Keypoints

Charles Zhao (hczhao), Anna Qin (alqin), Somya Arora (somyaa)

Background

Human pose estimation is an established and well-researched problem in the field of computer vision: given an input image containing one or more humans, detect their poses. This generally involves localizing specific keypoints, often the joints, on the human body; the keypoints may then be connected to model the position of the human body. Pose detection systems may focus on simply generating 2D models from 2D images, or they may use elements of 3D reconstruction to model the human in 3D. Furthermore, human pose estimation can be extended to pose tracking - analyzing motion and poses changing across time given an input video.

There are myriad applications for pose estimation systems. Robots may need to be built to recognize and react to certain gestures. Pose detection can help security in surveillance, analyzing the actions of those captured on screen. Autonomous driving systems benefit from more precise analysis of the actions of the pedestrians around it. Those in sports fields can evaluate whether their form is good or dangerous for their bodies, and in dance the accuracy of a given dancer's performance may be evaluated relative to a "correct" choreography.

Challenges involved in pose estimation include many of the common challenges faced by computer vision systems, such as variable lighting and color or occlusion, as well as more specific challenges that come about when trying to model a human body. Depending on the person's orientation relative to the camera, there may be occlusion of joints and body parts, requiring estimation from context. The system must be able to handle differences in appearance from person to person, due to different clothing, body types, etc. Some joints are particularly small and difficult to detect. Although human bodies follow the same general tree-like structure, there is still a high variability in how the body parts may be articulated, especially when flattened to a 2D image; on the other hand, there are plenty of poses that are physically impossible, and such predictions should somehow be filtered out of consideration.

Motivation

For the purposes of this project, we chose to focus on 2D, single-frame, single-person human pose estimation. In general, there have been two major approaches to this problem (Bulat and Tzimiropoulos, 2016). One strategy is to treat the location of a given joint as a continuous

variable to be predicted, and thus use regression to determine the coordinates of these keypoints.

The other common general approach is a detection-based approach, and involves overlaying the image with a grid and solving a classification problem for each coordinate: whether it contains the keypoint in question or not. A likelihood heatmap is generated for the image, and the location of the keypoint is set to be the location of maximum likelihood. In general, this approach has traditionally shown to give better results than pure regression (Sun et al., 2018).

However, detection strategies still have their limitations. Notably, heatmaps are limited by their resolution, so there are inevitable quantization errors because locations cannot be estimated continuously. Increasing the resolution of the heatmap mitigates this error, but doing so quickly becomes very computationally expensive. Additionally, the standard method of converting from heatmap probabilities to the location of a joint is to take the point with the maximum probability, but this max function is not differentiable, so the whole system cannot be trained end-to-end.

For our project, we wished to study the limitations of the detection-based approach, and investigate strategies that could potentially address these weaknesses. To do so, we looked to the regression-based approach for inspiration. By investigating this problem, we could gain a deeper understanding of the methodology of pose detection as well as the difficulties and limitations faced by existing strategies.

Related Work

Traditional pose estimation systems have relied upon pictorial structures or part-based models. In recent years, with the increasing research into convolutional neural networks (CNNs) applied to a variety of tasks, many human pose detection systems rely upon CNNs as well. With the success of DeepPose (Toshev and Szegedy, 2014), which formulated the pose estimation problem as a deep neural net regression problem with regards to body joints, such approaches gained popularity. Such systems have demonstrated strong performance that outstrips the performance of classical methods by a large margin.

In general, for 2D pose estimation, detection-based approaches have shown better performance than regression-based alternatives. Even a very basic network can be quickly trained to achieve respectable accuracy on detections in a relatively short amount of time (Xiao, Wu, and Wei, 2018).

One attempt to improve the performance is to construct a detection-followed-by-regression CNN cascade (Bulat and Tzimiropoulos, 2016). This method uses the detection approach to generate heatmaps, and then stacks these heatmaps to run them through a regression subnetwork, thereby naturally encoding context and part constraints.

Another approach uses integral regression to bridge the gap between regression and detection (Sun et al. 2018). After generating a heatmap for a given joint, the final joint location is

estimated by integrating over all locations in the domain weighted by their probabilities. Such a simple and lightweight approach, usable with any heatmap method, was in fact shown to give results comparable to state-of-the-art methods.

Design and Implementation

Base Heatmap Generation Model

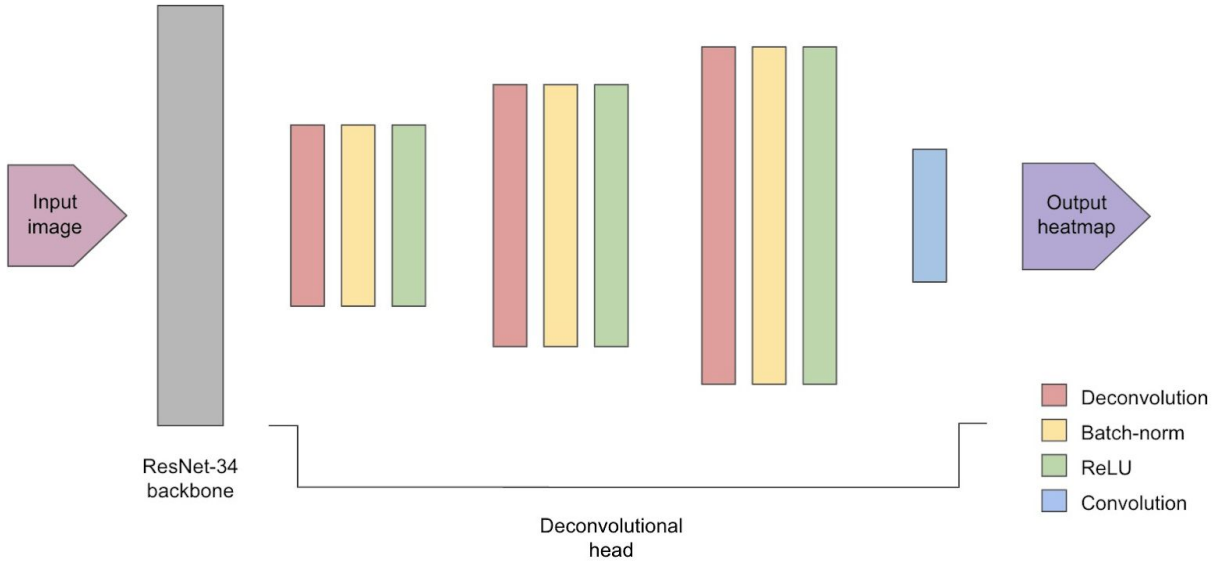
We ran our system using on Google CoLab using a Tesla P100-PCIE-16GB GPU.

We wished to build upon and refine an established neural network model. Initially, we attempted to use the Keypoint R-CNN model provided in the collection of torchvision models. This model was pre-trained on the COCO 2017 dataset and used a ResNet-50 backbone. It consisted of a region proposal network complemented by a series of alternating deconvolutional and ReLU layers for parts detection, as well as a convolutional layer. Testing was performed using the Leeds dataset (described below). After running the base system, we experimented with improving its base performance by adding and modifying layers to the system with the Leeds data.

We ultimately felt that this particular model did not serve as a good base to work off of for our project. After deciding to focus upon single-pose detection, it would have been inefficient to adapt the torchvision model to that purpose. Multi-person detection involves the usage of region proposal networks to create bounding boxes that locate the subject of interest. For the single-person Leeds dataset, we effectively just needed the “head” of this model, which is run on each proposed region. Although we decided not to continue working off this model, the experience and intuition we gained from working with it guided our later experimentation.

Therefore, we changed to the deconvolutional head model described by Xiao, Wu, and Wei (2018), which is similar to the above model, but without the network handling multi-person detection. ResNet, commonly used for image feature extraction, serves as the backbone. We have three deconvolutional layers with batch-normalization and ReLU, each with 256 filters and a 4x4 kernel and stride 2, then followed by a 1x1 convolutional layer to generate the heatmaps. The deconvolutional layers serve as upsampling steps to produce high-resolution feature maps, which Xiao et al. describe as crucial for good performance. This is one of the simplest structures that can be used to create heatmaps (1 heatmap per keypoint), and it already provides fairly good predictions.

Here is a diagram illustrating the basic structure:



Our code was adapted from pytorch-pose on GitHub (Yang). We used a pre-trained ResNet-34 backbone along with the 3-layer non-pretrained deconvolutional head, and trained this network ourselves on the Leeds dataset. We used a training-validation split of 11,000 to 1,000, and used the Adam optimizer with a learning rate of 0.0001. As in the original deconvolution head model, we used a mean squared error loss function on the heatmaps, with the target heatmaps being Gaussians of standard deviation 1 centered at each keypoint. We trained until the validation loss increased, at which point we know the model is beginning to overfit.

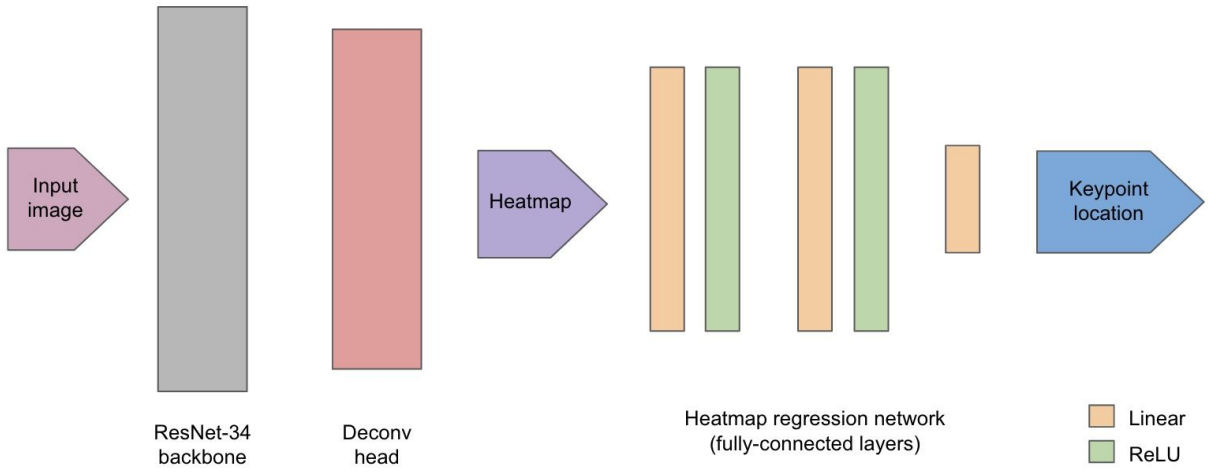
Our baseline heatmap-to-keypoint translation approach was to simply take the coordinate of maximum probability in the heatmap. Although this simple system performs fairly well, the limitations of this translation method, as discussed earlier, became apparent after our experimentation.

Heatmap Regression Network

Taking inspiration from the integral approach by Sun et al., our first approach to translating heatmaps to keypoint coordinates was to replace Sun et al.'s hard-coded soft-argmax function with a fully-connected neural network. We hoped that such a neural network would be able to learn a more complex, and better performing, mapping from heatmaps to coordinates.

Operating with limited computing power, we decided to investigate the capabilities of a very simple architecture. We built a 3-layer network for each joint (2 fully-connected layers of 128 neurons each, followed by a 2-neuron output layer) to take the heatmaps from the base model and output the predicted location of each actual keypoint. This way, the system might be able to learn patterns in uncertain heatmaps that would guide it toward finding the correct location.

Below is a diagram of the structure of our added layers:



Gaussian Mixture Model

We attempted another approach as well. One potential issue we saw with the integral regression method by Sun et al. was how it dealt with multi-modal heatmaps, or with how it dealt with uncertain heatmaps in general. Since it performs an integral over the whole heatmap, a multi-modal heatmap would result in this method predicting somewhere in between the modes.

Consider the following predicted heatmap output by the base model:

Here, both knees are seen as likely candidates for the right knee keypoint. This sort of bimodal distribution has a tendency to arise because of the symmetric nature of the human body.

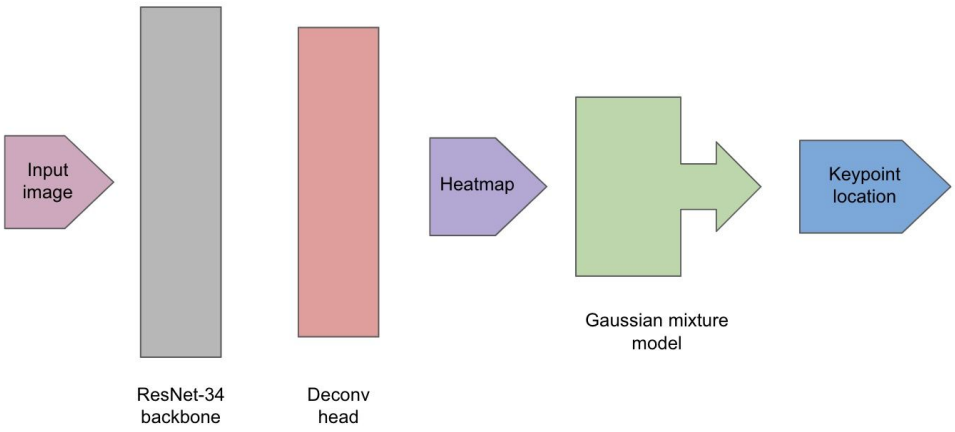
The basic method of mapping from a heatmap to an actual location prediction is simply by taking the point with the maximum probability. However, this suffers from the aforementioned drawback of quantization, and may ignore other useful information provided by the heatmap. Furthermore, we believed that in such cases where there are multiple modes of high probability, the integral approach would perform poorly. Therefore, we hypothesized that a more tailored strategy of conversion, using a Gaussian mixture model (GMM), may address this weakness.

Some benefits of this approach are that it does not require any additional training, and that it is general in that it can be run on top of any keypoint heatmap generator.

The Gaussian mixture model assumes that all data points are generated from a mixture of Gaussian distributions with unknown parameters. If our system detects multiple candidates for a keypoint location, this approach would be able to distinguish them and only use the part of the heatmap relevant to the most likely candidate. This approach incorporates more of the heatmap results into the prediction, while mitigating skewing due to outliers or multiple modes.

We select the 20 points with the highest probability from the heatmap and cluster them using a GMM. The GMM fits three Gaussians to this sample using expectation maximization, weighting each Gaussian according to its contribution. We select the Gaussian with the greatest weight, and take its mean to be the keypoint location. We use the Bayesian Gaussian Mixture Model provided by Scikit-Learn (Pedregosa et al., 2011).

For comparison with the previous, here is a diagram illustrating this approach:



Deduplication

After building this system, we observed that when the base model is uncertain about which side is left and which is right, it sometimes puts both the left and right keypoints at the same location (i.e. both heatmaps have the same argmax). Therefore, we came up with an approach building off of our GMM approach that explicitly tries to prevent this scenario. The algorithm is as follows:

1. As in our previous approach, fit a mixture of three Gaussians for each keypoint heatmap.
2. Sort Gaussians for all keypoints by weight, in decreasing order.
3. For each Gaussian:
    a. If we've already accepted a coordinate for this keypoint, continue to next Gaussian.
    b. If this is not a lateral keypoint, continue to next Gaussian.
    c. If weight of this Gaussian < 0.01, continue to next Gaussian.

    d. If this Gaussian's mean is within sqrt(5) pixels of an already accepted coordinate, continue to next Gaussian.
    e. Accept this Gaussian's mean as the coordinate for this keypoint.
4. For any keypoint that still has not been assigned a coordinate, assign it the coordinate of the Gaussian with the greatest weight.

In summary, our deduplication GMM approach uses the second or third fitted Gaussians as alternatives to the first to prevent lateral keypoints from being predicted to be at the exact same location.

Dataset

The Leeds sports dataset, combining the original and extended versions, features 12,000 pose annotated images, each cropped and centered around a single person, scaled so that the person is about 150 pixels in length (Johnson and Everingham, 2011). The images were gathered from Flickr using sports-related search terms such as 'athletics,' 'parkour,' and 'gymnastics.' When fed into our model, the images are first resized and padded to 256x256 pixels. Each image in the dataset has been annotated by Mechanical Turk workers with 16 joint locations, with right and left labeled from the point of view of the person. A joint is stored with its x and y location, and a binary value indicating whether or not it is visible in the image.

Here is a sampling of images, without their keypoints:

Because the majority of the images are of people performing sports activities, there is a large variety of highly articulated poses. Exposure to a greater range of human poses will help the network become more robust in analyzing new poses. However, because many of the samples in the dataset feature individuals wearing sports uniforms, they may be limited in the variability in appearance that different clothing can cause.

A potential drawback of this dataset is its limited size and variety, especially in comparison to other more expansive datasets such as MPII or COCO. Naturally, a neural net will be able to learn more and perform better if it is trained upon a wider diversity of data. However, focusing on this dataset was sufficient for the scope of model capability we wished to explore.

Results and Analysis

Evaluation Metrics

Quantitative

We measured the accuracy of our system by calculating the percentage of correct keypoints (PCK), which is calculated by dividing the number of correctly predicted keypoints by the total number of keypoints. A keypoint is considered correctly detected if the predicted keypoint location is within a normalized threshold distance of the true location.

For training our regression network, as well as a general performance metric, we used the mean squared error: the square of the Euclidean pixel distance between the predicted and the true keypoint locations.

We also recorded the time required during evaluation for each of our approaches. As there is often a tradeoff between accurate results and the amount of time needed to reach said results, we wanted to see what kinds of models could achieve reasonable results in a reasonable timeframe.

Qualitative

In addition to the above, we made qualitative judgments of the successes and failures of our system to guide our experimentation.

The heatmaps generated by our base network were visually examined; blue-range colors indicate low probability of the keypoint being located there, while warm-colors indicate high probability of the keypoint location. By observing the patterns of confidence for various keypoints, we were able to learn more about how our system made its predictions.

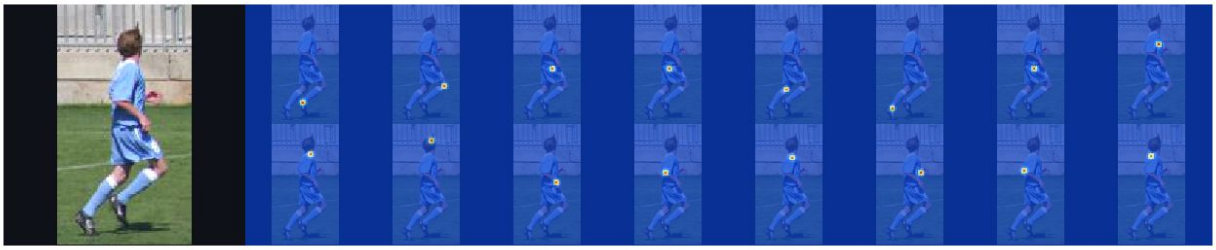
By nature, the heatmaps encode more information as to how the system arrives at its final predictions. However, for proper end-to-end comparison, we also needed to examine the pure keypoint predictions. We implemented a function to display the keypoints atop the image and draw lines between the keypoints for a better stick-figure visualization of the body pose that is predicted.

Baseline Model Performance

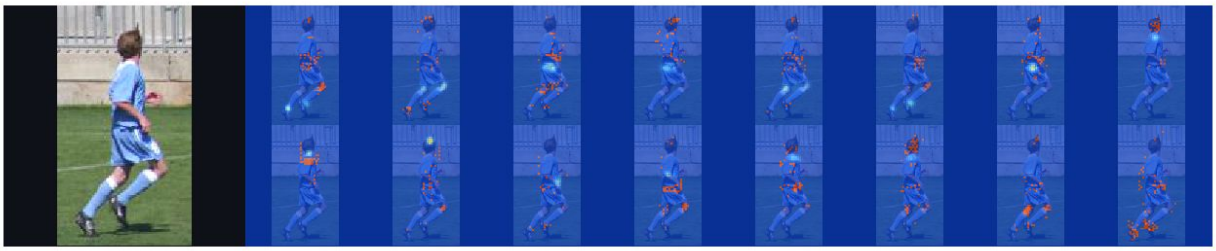
Our baseline model, which simply takes the argmax of each heatmap, had the following performance on the test set:

- PCK: 0.662
- MSE: 33.2
- Time per image (ms): 55.6

For qualitative context, here are the ground truth keypoints for one example:



And here are the heatmaps predicted by the simple deconvolutional head:



Each keypoint location was predicted individually. Below, we highlight some example heatmap predictions for comparison. The ground truth location is pictured on the left, and heatmap on the right.

Some keypoint predictions were relatively good, high confidence in the correct location:

Some were poor, showing high confidence in an incorrect point (here, the left and right feet were mixed up):

or low confidence all around:

And some were in between, showing moderate confidence in a correct general area:

We can also compare their poses directly, by drawing the predicted keypoints on the image and connecting them:

Ground truth Prediction

Even the baseline model already gives us fairly strong performance. With this particular example, the clearest apparent error it makes is switching the left and right ankles.

Heatmap Regression Network Performance

- PCK: 0.075
- MSE: 274
- Time per image (ms): 65.1

Unfortunately, we were not able to find a well-performing fully-connected architecture. All architectures we tested, varying the number of layers and neurons per layer, performed significantly worse than the max method. Training the network for more epochs did not appear to help the performance, as it appeared to quickly get stuck at a poor local minimum.

We attempted both to freeze the weights of the heatmap detection network, and to allow them to be modified in a unified end-to-end training process. We found that keeping the original

heatmap frozen gave moderately better results. However, this may simply be because the added regression network was performing so poorly.

Note that some of the keypoints are not even predicted within the image.

Due to the nature of neural networks, it is difficult to analyze where they went wrong. We hypothesize that this system failed because of the difficulty of learning and preserving spatial context through linear, fully-connected layers. This, for example, is why most networks working with images utilize convolutional layers, but a strictly convolutional layer approach would only give us another 2D representation, not a coordinate vector like we desire. Further investigation would be necessary to determine a more sophisticated network architecture.

Gaussian Mixture Model Performance

- PCK: 0.645
- MSE: 30.4
- Time per image (ms): 463

In order to qualitatively look at the performance of our GMMs, we visualize the generated clusters on top of the heatmap. Here are two examples, the first for the right knee, the second for the left ankle.

Our Gaussian mixture model approaches gave slightly better MSE than the max approach, presumably due to taking into account more of the heatmap and overcoming the quantization issue. However, it is apparent that for many images, the simple intuitive method of finding the maximum probability is sufficient for prediction; it is likely that the GMM process often ends up selecting the approximate overall maximum anyways. As expected, the process of clustering and fitting the model also adds a significant amount of computational time during evaluation, as it is a more complex operation than taking the maximum.

Deduplicated GMM

- PCK: 0.638
- MSE: 31.3
- Time per image (ms): 443

The deduplicated GMM approach did perform better on certain problematic images. In particular, it can disambiguate left and right keypoints when these keypoints' heatmaps have the same argmax, as we had hypothesized. In the example below, the max approach predicts the left and right ankles to be at the same location, whereas the deduplicated GMM approach gives a more accurate prediction of the locations of both ankles.

Ground Truth

Max

Dedup GMM

Performance Summary

We summarize accuracy (PCK), loss (MSE), and time elapsed per image for our different approaches of heatmap-to-location translation in the following table:

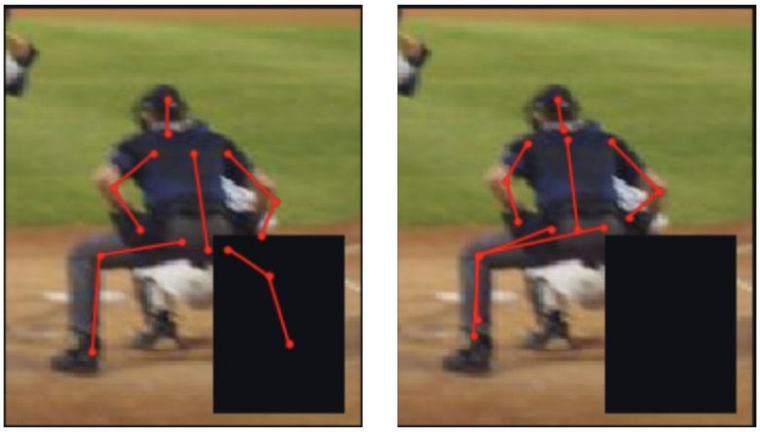
	PCK	MSE	Time/Image (ms)
Max	**0.662**	33.2	**55.6**
FC	0.075	274	65.1
GMM	0.645	**30.4**	463
Dedup GMM	0.638	31.3	443

Occlusion Experiments

We used multiple approaches to analyze the performance of our systems when faced with occlusion, a common difficulty in pose estimation. Our occlusion experiments were meant specifically to test the performance of our models under uncertainty.

In the first approach, we implemented a random erasing transformation throughout the entire dataset. This transformation created random sized rectangles that covered 2% to 40% of the image. The coordinates, size and aspect ratio were chosen randomly for each image. The pixels that were part of the randomly generated rectangle were zeroed out. These emulated what occluding objects could look like in real life.

Here is an example occlusion. On the left is the ground truth; the right leg is now hidden behind a zeroed-out section of the image. On the right is the prediction generated by our GMM model (without deduplication; notice that overlap of predicted location for the knees and feet).



We applied the transformation to the dataset and compared the evaluation metrics (accuracy, loss) across the four approaches:

	PCK	MSE	Time/Image (ms)
Max	**0.416**	116.4	**54.1**
FC	0.027	330.7	67.9
GMM	0.406	**95.8**	221.5
Dedup GMM	0.372	106.6	244.7

The max, GMM, and dedup GMM approaches all demonstrate similar performance. However, note that the two GMM approaches show a lower PCK but a lower MSE as well. This suggests that although the GMM methods may have slightly fewer predictions within the threshold of the truth, their predictions in general are closer to the truth, i.e. they are more precise. This makes intuitive sense, because one goal of our GMM strategies was to produce more precise predictions by overcoming quantization error and incorporating more of the heatmap information into the coordinate translation.

After implementing random occlusions, we were curious to test the performance of our models under occlusion of different parts of the human pose. We applied a targeted occlusion transformation, where we targeted either the upper body or the lower body to be zeroed out. (“Upper PCK” refers to PCK when the upper half is occluded, etc.)

	Upper PCK	Lower PCK	Upper MSE	Lower MSE
Max	**0.18**	0.30	246.0	184.8
FC	0.05	0.02	1196.0	418.3
GMM	0.17	**0.33**	207.2	161.2
Dedup GMM	0.16	0.32	**205.0**	**160.1**

Overall, the systems show significantly lower loss and higher accuracy when the lower half is occluded, suggesting (as expected) that the upper part of the human is more important to correctly determining the pose.

As we had hypothesized, the GMM approaches tend to perform better than the max approach under uncertainty, because it draws upon more of the data encoded in the heatmap. However, this improvement is quite minor and may not warrant the extra evaluation time needed.

Conclusions and Further Work

We gained valuable insight from each of our experiments. A pure fully-connected neural network approach to heatmap-to-keypoint translation seems unlikely to perform better than tailored problem-specific approaches. The spatial aspect of the problem should somehow be encoded in the model.

GMMs can overcome the quantization issue of the max approach by using more of the heatmap than just the max, while performing well under uncertainty by only focusing on relevant parts of the heatmap. The former was demonstrated by their better MSE performance on our original test set, and the latter was demonstrated by their better performance on our occluded test set.

Furthermore, using deduplicated GMMs can reduce pose detection errors in which two keypoints are predicted to be at the exact same location. However, since this only affects a very specific subset of images, the deduplicated GMM approach did not perform significantly different from the original GMM approach on the test set as a whole.

As mentioned previously, the Leeds sports dataset is somewhat limited in its scope. It would be worth evaluating the performance of our systems after training upon more diverse and expansive data, and see what other situations are handled well or poorly. Such an exploration would likely grant more inspiration as to other ways to improve the system. For example, the deduplication rule we implemented was in direct response to an observation we had after examining our predictions; there may be other such guidelines we can impose or encode into a network to support the core prediction model.

There are pose datasets that take advantage of more keypoints on the human body; this information may improve performance if leveraged correctly. Our pose estimation approach only looks at joints on the human body, but perhaps if used in conjunction with a segmentation system or some other model that can examine the limbs on the human body as well, then that could improve performance.

Our current systems individually, independently predict the location of each keypoint; one entirely separate heatmap is created per joint. However, it is possible that letting information be shared between the different keypoint detections may allow for better results given increased contextual information. This, of course, requires a more complex model to be used.

Our findings suggest that a Gaussian mixture model approach may be useful in generating more realistic pose estimations that, for example, try not to predict two different keypoints to be at the same location. One potential direction for future study would be a hybrid of the deduplicating GMM approach and either the max approach or the integral approach. For

example, one could use our deduplicating GMM to propose a region of interest for each joint, and then find the argmax or the integral on these regions only rather than the whole heatmap.

Our heatmap regression network performed poorly, but we still feel that there is potential for a neural network to perform well with this task. After all, the general trend in recent years has been away from part-based models and toward increased neural network usage. Given more time and computing power, as well as some deeper insight as to the capabilities and characteristics of neural nets, a network architecture that is better at learning for this particular task could likely be developed.

Acknowledgements

We would like to thank our project advisor Fangyin Wei for her guidance, as well as Professor Olga Russakovsky and Felix Yu for their advice and feedback.

References

Bulat, A., & Tzimiropoulos, G. (2016). Human Pose Estimation via Convolutional Part Heatmap Regression. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 717–732.

Johnson, S. & Everingham, M. (2011). Learning Effective Human Pose Estimation from Inaccurate Annotation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2011)*

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, *JMLR 12*, 2825-2830.

Sun, X., Xiao, B., Wei, F., Liang, S., & Wei, Y. (2018). Integral Human Pose Regression. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, 536–553.

Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, B., Wu, H., & Wei, Y. (2018). Simple Baselines for Human Pose Estimation and Tracking. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, 472–487.

Yang, W. (n.d.). Retrieved from https://github.com/bearpaw/pytorch-pose

Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. arXiv preprint arXiv: 1708.04896 (2017)

Links

Dataset:

https://sam.johnson.io/research/lsp.html

https://sam.johnson.io/research/lspet.html

DeconvHead implementation:

https://github.com/bearpaw/pytorch-pose