Team Skeletal Insights

FINAL PROGRESS REPORT - Preliminary Results

Data Collection

We built a web-scraper to pull yoga images from Google images using search words. To build our yoga corpus and to limit the amount of time we had to spend tagging, we only scraped for one pose at a time - using the search term as the pose label for the resulting scraped images. We began using English terms because we believed that they would return the largest number of results and we were worried about the Sanskrit terms having a low frequency of usage. Our prediction on this score was incorrect. While the English terms did return the largest number of results, the results were incredibly messy and the amount of usable images was very low for poses (other than downward dog and warrior 2). Searches using the Sanskrit terms returned cleaner results and a decent number of images. Web scraping for each of the Sanskrit pose names returned about 400 images per term and about 200 were usable.

In order to have a fully-functioning yoga learning application, we need to build a model that is not only capable of recognizing poses which differ greatly from one another, but could also differentiate poses which only differed slightly from each other. Therefore, we identified three pose groups (standing, inverted, and seated) and picked a couple of poses for each group.

STANDING POSES

Virabhadrasana I (warrior 1)

Virabhadrasana II (warrior 2)

Virabhadrasana III (warrior 3)

Namaskara Natarajasana (lord of dance pose hands in prayer)

Nantum Natarajasana (bowing with respect lord of the dance pose)

INVERTED POSES

Ado mukha svanasana (downward dog)

Hastapaadasana (standing forward bend)

Salamba sarvangasana (shoulder stand)

Salamba sirsasana (supported headstand)

SEATED POSES

Gomukhasana (cow-faced pose)

Bharadvajasana (Buradvaja’s twist)

Pasasana (Noose)

Even though the Sanskrit web-scraping returned cleaner results, some data cleaning was still necessary. An image was removed from the corpus if it had any of the following attributes:

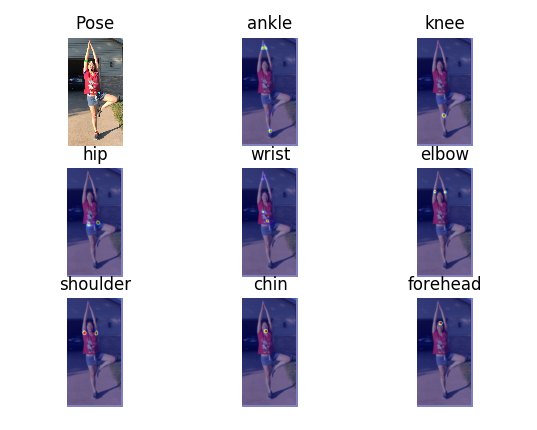
* 1. Multiple people
  2. Body parts that were cut-off or obstructed
  3. Non-human
  4. Poor resolution and/or too small

These data collection and cleaning methods leave us with a sufficient dataset to start running classification models. However, we would like to build an image-to-label neural net, for which we will require a larger dataset. To supplement our web-scraping results, we plan to apply artificial noise to our current labeled dataset to increase the size of our corpus.

**Preliminary Results**

Step 1: Joint Coordinates

Once we had created our labeled dataset, we could begin to turn these images into vectors of joint locations. Initially, we thought that it would be necessary to translate joint coordinates into joint angles. However, the neural nets that we were utilizing were trained on non-inverted poses. This training bias resulted in the occasional difficulty distinguishing between an ankle and a wrist - potentially resulting in unrealistic joint angles. Since the main thing that we cared about was the relationship between joints, we decided to just stick with using joint coordinates rather than angles. The neural net models that we explored were DeepPose and DeeperCut. Eventually, we settled on the DeeperCut model due to its speed and confidence in the coordinates it produces.[[1]](#footnote-0)



*Image 1: Sample visualization of DeeperCut algorithm in recognizing joint coordinates*

Step 2: Binary Classification

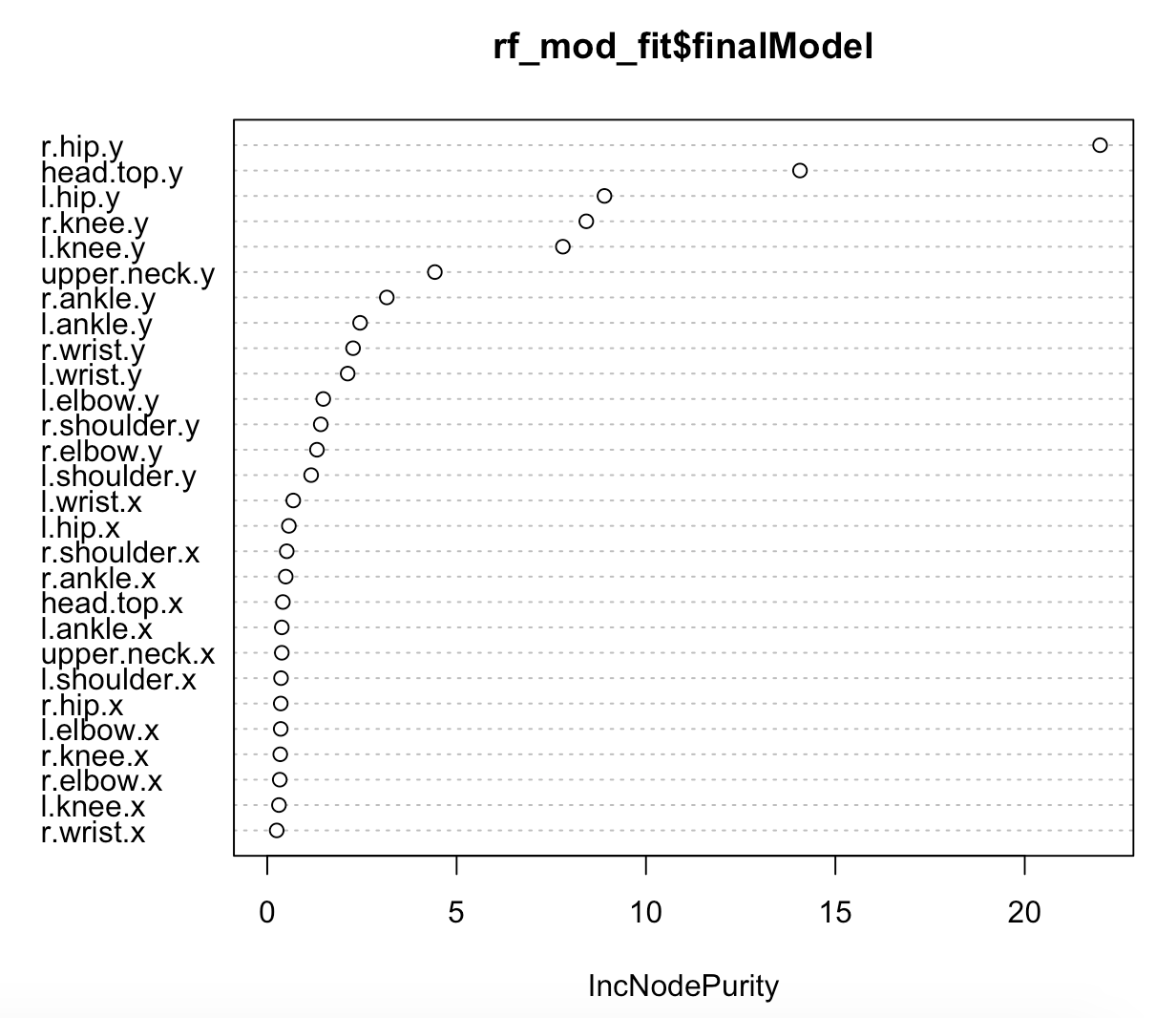
In order to make sure that using only joint coordinates does not jeopardize our classification attempts, we built a few binary classification models. The purpose of these binary classification models is twofold. Firstly, we wanted to make sure that the coordinates being produced from the neural nets run on 2D images could actually be used for classification (e.g. sanity-checking our DeeperCut results). Mostly crucially, we needed to see if we needed to add a yoga-specific layer to the DeeperCut model to get usable results. Secondly, how the binary models run might give us clues about which multi-class models we should explore. We had an 80-20 training-test split and ran a 5-fold cross validation using logistic regression, random forests, and support vector machines.[[2]](#footnote-1)

Since we are eventually going to be using multi-class classification models, we decided that it was not worth it to check all of the possible binary combinations of poses for classification. To narrow our focus, we decided that we needed to check whether joint coordinates could be used to learning big differences (e.g., downward dog versus warrior pose) as well as small differences (e.g. warrior 2 versus warrior 3). We built models looking at three poses: downward dog, warrior 2, and warrior 3. As mentioned in the Methodology section, we will be comparing three models: logistic regression, random forest, and svm.

1. Downward Dog versus Warrior 2

For logistic regression, the final model return by cross validation highlighted the following joints as being significant indicators of downward dog versus warrior 2: right ankle, right knee, right hip, left knee, left ankle, and head. In particular, the model cares about the y coordinates for the head, right ankle, right knee and right hip and the x coordinates for the left ankle and left knee. It would appear that the model learns whether the pose is inverted (a clear indicator of downward dog) and whether one leg is extended out or not (a clear indicator of warrior pose). Logistic regression does a reasonably good job of classifying poses. At a threshold around 0.5, the accuracy rate is 88%.

For random forests, with a threshold of 0.5, the accuracy rate was a little higher at 90%. If we look at the plot of variable importance (Plot 1), we can see that, for the most part, the random forest model is paying attention to the same joints that the logistic regression model is. However, the random forest model focuses primarily on the y coordinates rather than the x coordinates and also pays more attention to the arms (whereas the logistic model does not pay attention to the arms at all). SVM models are difficult to interpret which variables are contributing the most. The accuracy rate for the SVM model is 92%.

**Plot 1: Random Forest Variable Importance (Downward Dog vs Warrior 2)**

In order to select the best model for this classification task, we can look at a couple of metrics: accuracy rate, R-squared, RMSE, and AUC. It might be tempting to simply choose based on accuracy scores, but since we randomly selected the threshold we will probably want to use a few additional metrics. These metrics can be seen in Table 1 for each model. Logistic regression does the worst across the board. SVM and Random forest appear to be tied.

**Table 1. Downward Dog versus Warrior 2 Model results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **R-squared** | **RMSE** | **AUC** |
| **Logistic Regression** | 0.88 | 0.67 | 0.29 | 0.95 |
| **Random Forests** | 0.90 | 0.78 | 0.24 | 0.96 |
| **SVM** | 0.92 | 0.75 | 0.25 | 0.97 |

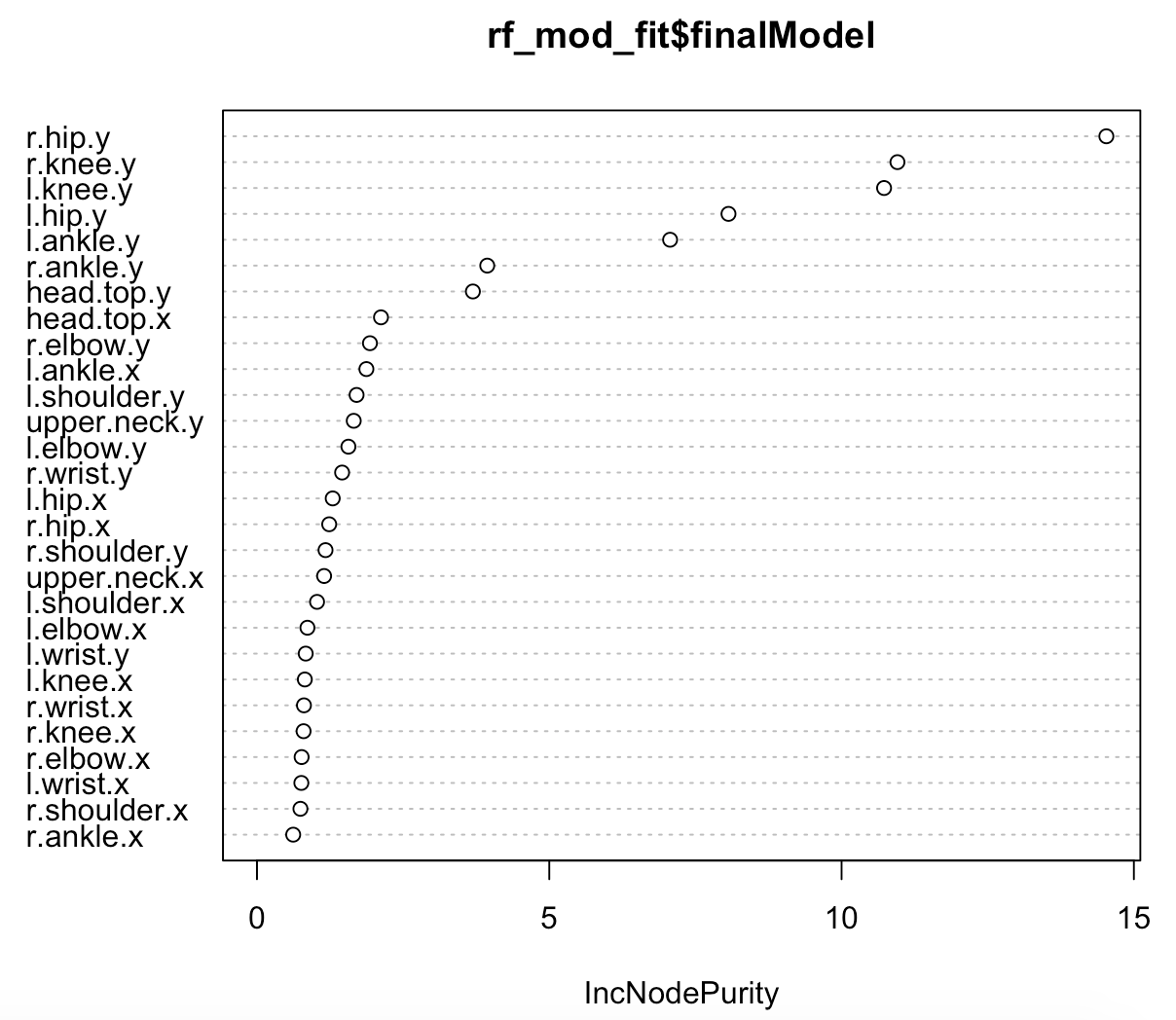
1. Warrior 2 versus Warrior 3

Across all of the possible metrics, the three models did a worse job classifying two poses that were similar (which is unsurprising because it is a more difficult problem). The accuracy rates are still reasonably good, so we can conclude that joint locations are sufficient for classification. For these two poses, random forest is clearly the winning model as it does the best for most of the metrics.

**Table 2. Warrior 2 versus Warrior 3 Model Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **R-squared** | **RMSE** | **AUC** |
| **Logistic Regression** | 0.88 | 0.52 | 0.34 | 0.96 |
| **Random Forest** | 0.92 | 0.65 | 0.30 | 0.98 |
| **SVM** | 0.91 | 0.48 | 0.38 | 0.96 |

Since the random forest model is clearly the model that works the best for both small and large differences, we should look to see if the variable importance is the same for big and small differences. For these poses, the head coordinates are not as important. However, similar to the other random forest model, the y coordinates appear to be more informative than the x coordinates. This model does pay more attention to hip position than the previous one.

**Plot 2: Random Forest Variable Importance (Warrior 2 vs Warrior 3)**

**Next/Remaining Steps**

1. Multi-class classification models on DeeperCut results
2. Add artificial noise to yoga corpus
3. Image-to-label neural net

1. We did not train the joint coordinate model using our yoga data since we currently do not have a labeled training set for the joints in those images (see more information on this later). [↑](#footnote-ref-0)
2. This paragraph belongs in the Methodology section, but is included here temporarily [↑](#footnote-ref-1)