CSCI5561 Final Report: OpenMonkey Challenge

Archit Kalla
kalla100@umn.edu

Jane Huynh
huynh369@umn.edu

Vishwajith Kaushik Ramesh
rames115@umn.edu

Collin Henderson
hende906@umn.edu
University of Minnesota - Twin Cities
Minneapolis, MN

Abstract

The OpenMoneky Challenge is an open challenge where competitors can submit their solutions to the problem of non-human primate (NHP) pose estimation. In this paper, we discuss the existing work on the challenge and multiple implementations of NHP pose estimation. In our solution, we used DeepLabCut to get the pose estimation of the primates and YOLOv5 to perform monkey species classification. In our results, our monkey family specific feature extraction models performed better than our general monkey feature extractor with smaller average mean per joint position error.

1. Introduction

> Non-human primates remain of great interest in biomedicine and related fields, including in neuroscience and psychology, as well as in anthropology, epidemiology, and ecology. Automated tracking can also benefit animal welfare programs, veterinary medical practice [4], and, indeed, conservation projects [14]

There are very important uses of motion capture [13] and tracking in the veterinary field to identify potential pose variations that may lead to early onset diseases. Another important facet of working with pose estimation on primates is to identify which species the primate belongs to. This, in application, can be used to better understand any variances in the pose of each primate.

2. Related Work

Deep High-Resolution Representation Learning for Visual Recognition [19]: High-resolution representations are essential for position-sensitive vision problems like human pose estimation. Deep convolutional neural networks (DCNNs) can learn richer representations, owing to this advantage they are widely used in many computer vision tasks such as object detection and human post-estimation. Some of the popular DCNN frameworks such as GoogleNet and ResNet first encode the input image as a low-resolution representation and then recover a high-resolution representation from them [6]. HRNet on the other hand maintains the high-resolution representations throughout, resulting in a semantically richer and spatially more precise representation. In the network high-to-low convolution streams are connected in parallel. The image is processed by a stem, which consists of two stride-2 (3 × 3) convolutions decreasing the resolution to 1/4 , and subsequently the main body that outputs the representation with the same resolution (1/4). By maintaining high-resolution representations the network generates reliable high-resolution representations with strong position sensitivity through repeatedly fusing the representations from multi-resolution streams. [19]

Human pose estimation using HRNet aims to detect the locations of landmarks on the human body (e.g., elbow, wrist, etc) from an image. The framework transforms this problem to estimating K heatmaps of size W/4 × H/4 {H1, H2, ... , HK }, where each heatmap Hk is the confidence of locality of the kth landmark. The heatmaps are then regressed over the high-resolution representation output. The loss function, defined as the mean squared error, is applied to compare predicted heatmaps to the ground truth ones. Some example results are presented in figure 2 [19].

Multi-animal pose estimation and tracking with DeepLabCut [12]: DeepLabCut is a popular open-source pose estimation toolbox that provides high-performance animal assembly and tracking—features required for robust multi-animal scenarios. In this paper to extend the open-

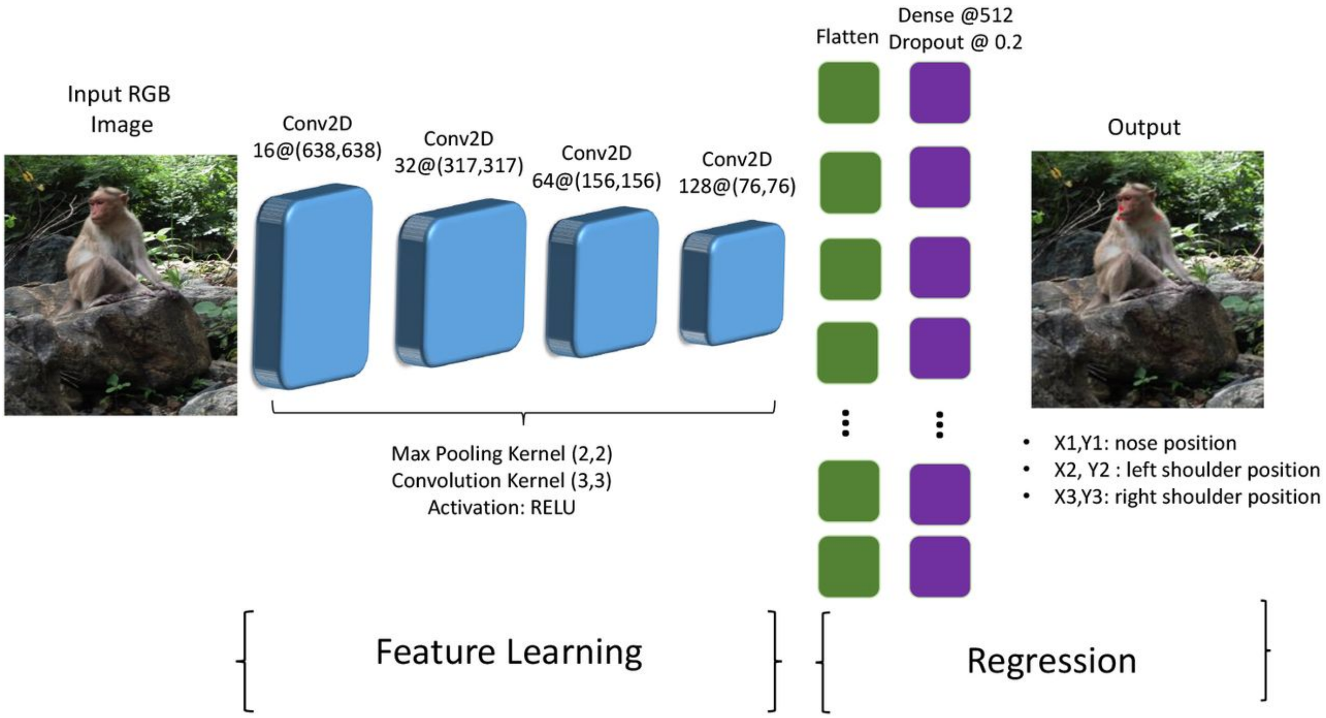


Figure 1. Feature extraction using deep CNN network [11]

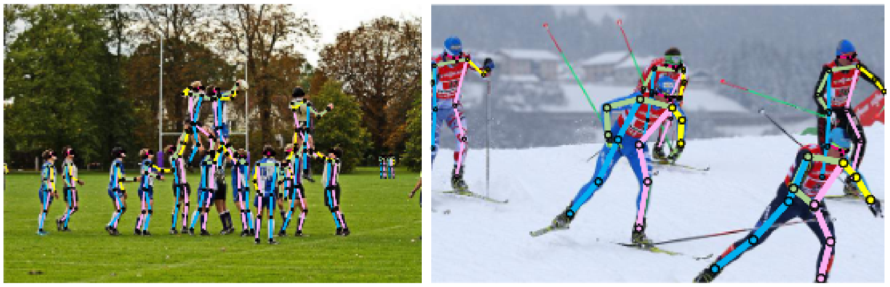


Figure 2. Qualitative human pose estimation results over representative images (arXiv:1908.07919)

source DeepLabCut software to multi-animal scenarios, they designed a practical and almost entirely data-driven solution. The larger goal was broken down into smaller sub-tasks: keypoint estimation, animal assembly, local tracking, and global "tracklet" stitching. The framework was tested with four databases of varying complexities and was released to serve as a benchmark for future algorithm development.

The ability to quickly perform species classification using images is also of high interest to research communities. In general, classification is done through image mining where we extract information from images such as sharp edges, color, segmentation, etc [15]. A commonly used technique to perform classification is transfer learning where information learned from pretrained models can be applied for new tasks [20]. A paper from 2019 applied this technique using a pretrained Mask R-CNN model and Inception Nets to train a bird species classification model. The resulting model had an F1 score of 55.67% [9]. Another paper had applied this technique with a pretrained VGG-19 model to perform tree species identification. The resulting model had an accuracy 99.70%. [17] A more recent paper, published in 2021, used YOLOv5 to perform flower species classification. This had a higher mean average precision compared to Faster R-CNN and YOLOv3. [18] Since its release in 2020, YOLOv5 has gained high popularity due to its high precision and training speed compared to existing computer vision models such as mask R-CNN [3]. Mask R-

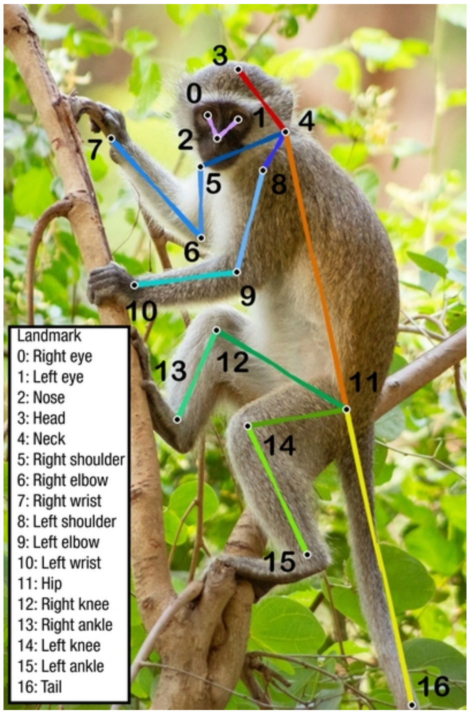


Figure 3. Example of the output of the pose estimation and extracted features

CNN is an extension from fast R-CNN developed in 2017 that detects objects and generates segmentation masks in parallel [5]. In a previous study to detect fish heads and tails, YOLOv5 had performed better than mask R-CNN in precision, recall, and mAP [16].

3. Baseline Method

For our project, we will base our methodology on work done for the "Monkey Features Location Identification Using Convolutional Neural Networks" research paper. [11] In this study, researchers had used a deep convolutional neural network (CNN) to detect monkey features from RGB images. These images, similar to our given training dataset from the Open Monkey Challenge, were of monkeys in natural environments. Their CNN model learned monkey features during the training process using convolutional networks. Their network included 4 convolutional layers where the filter size doubled per each layer with the first being size 16. The network then uses a max pooling kernel, convolutional kernel, and ReLU activation function to extract image feature maps. They also had 20% dropout and calculated loss on the training dataset using Mean Squared Error. For faster computation, the images were also resized to 640x640 pixels. We also plan on comparing our results against DeepLabCut, which allows 2D and 3D pose estimation of animals using models that are already trained.

4. Proposed Method

We propose a top down approach where we identify monkeys in the image and estimate joint pose within determined bounding boxes. In addition to this, we will identify the monkey family the primate in the image belongs to. To determine monkey bounding boxes and determine the monkey family, we will use transfer learning with a pretrained YOLOv5s model and YOLOv5 to train a monkey species classification model. The pretrained model and code for YOLOv5 can be found here: https://github.com/ultralytics/yolov5. For the pose estimation we will use DeepLabCut, an animal pose estimation tool to identify the poses of each primate. The code for DeepLabCut can be found here: https://github.com/DeepLabCut/DeepLabCut. We will need to feed the training data from the OpenMonkey challenge website to DeepLabCut and then test/validate as well on the given data. We will train separate models for new world monkey, old world monkeys, apes, and all monkeys. Afterwards, we will use the generated models and run them on the respective challenge images identified by the classification model. This method differs from other methods as instead of just identifying the pose estimation, we plan to take it a step further in identifying primate type. This additional information can prove useful in practice as having a primate type can help diagnose medical conditions appropriately. The full image dataset used for training can be found here: https://competitions.codalab.org/competitions/34342.

5. Evaluation

Following submission guidelines for the Open Monkey Challenge, our work will be evaluated using 3 metrics. The first metric is Mean per Joint Position Error which is expressed as:

$$\mathrm{MPJPE}_i = \frac{1}{J}\sum_{j=1}^{J}\frac{\|\widehat{\mathbf{x}}_{ij}-\mathbf{x}_{ij}\|}{W} \tag{1}$$

In essence, Eq. (1) is the average euclidean distance between the ground truth point and our models predicted point of a joint. The next metric is the Probability of correct keypoint which is broadly defined as detection accuracy given error tolerance, where the larger value indicates better performance. This can be expressed as:

$$\mathrm{PCK}@\epsilon = \frac{1}{17J}\sum_{j=1}^{J}\sum_{i=1}^{17}\delta\left(\frac{\|\widehat{\mathbf{x}}_{ij}-\mathbf{x}_{ij}\|}{W}<\epsilon\right) \tag{2}$$

Finally we will use Average precision which can be defined as:

$$\mathrm{AP}@\epsilon = \frac{1}{17J}\sum_{j=1}^{J}\sum_{i=1}^{17}\delta(\mathrm{OKS}_{ij}\geq\epsilon) \tag{3}$$

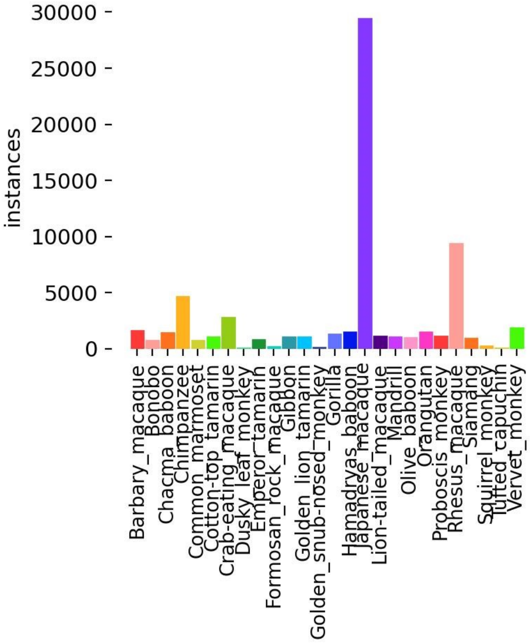


Figure 4. Label distribution of training data

Where OKS is defined as:

$$\text{OKS}_{ij} = \exp\left(-\frac{\|\widehat{\mathbf{x}}_{ij} - \mathbf{x}_{ij}\|^2}{2W^2k_i^2}\right) \tag{4}$$

The OKS is a similarity measure between two points where k is the tolerance per landmark. The code that will evaluate the performance of our results can be found here https://github.com/yaoxx340/MonkeyDataset. Our goal is to either have better performance in at least one of these metrics against a baseline implementation using DeepLabCut.

6. Results

6.1. Classification

A 26 species monkey classifier model was trained using the full training dataset provided by Open Monkey Challenge using transfer learning. To train this model, YOLOv5 was used with a pretrained YOLOv5s model with the initial 10 backbone layers frozen to speed up training. Training was performed for 50 epochs with a batch size of 32 and took 49 hours to complete. The resulting confusion matrix is depicted in figure 5. This matrix had strong true positive diagonal values overall although a few species had a low percentage of true positives due to the extremely uneven distribution of label data as seen in figure 4. The overall validation accuracy was 75.60% with a mean average precision @ 0.5 IoU tolerance (mAP50) score of 79.50%. Note that this value and the confusion matrix, especially for values along background labels, appears worse than the true values due to insufficient labels on validation data. In the case where there are multiple monkeys in an image, labels would only be given for a single monkey. After performing inference on the validation data, images were sorted based on the monkey family of the identified monkey species. In the case of multiple monkeys being identified, the monkey family was chosen based on the highest confidence score. In the case where no monkeys were identified in an image (0.3% of validation images), a random monkey family was selected. The determined monkey family validation accuracy from this process was 98.71%. Code required to run YOLOv5 with Open Monkey dataset and fully trained model can be found here: https://drive.google.com/file/d/1AxplRSGaKQtXphwmTLf3jCeK96dEXV-U/view?usp=sharing/

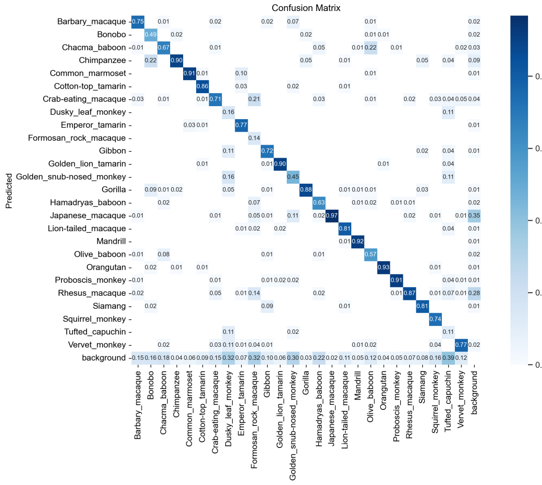


Figure 5. Confusion matrix for monkey species classification

6.2. Feature Extraction

The method in which we extract features for landmark detection remains as mentioned in Section 4. We used DeepLabCut to train separate models for each species family in hopes of getting better performance as evaluated in Section 5. The training dataset has over 66 thousand images to train over, however there was quite a few examples where there were multiple primates in the same image as well as in the same image and bounding box as well. Due to some limitations in the DeepLabCut library we were unable to crop the images and maintain the ground truth landmarks given to us in the training data. However this did not seem to be an issue due to the fact that when training the models we were giving precise coordinates where the joints in question were located. Image augmentation was done automatically in DeepLabCut when extracting frames for the training which increases the model's resilience to color and rotational changes in test data. The following tests were run on the validation data available on the OpenMonkeyChallenge Website as the test data was unavailable at this time.

6.2.1 New World and Old World Primates

This section will explore the development of the models used for the post estimation of Old world and New world monkeys [1].

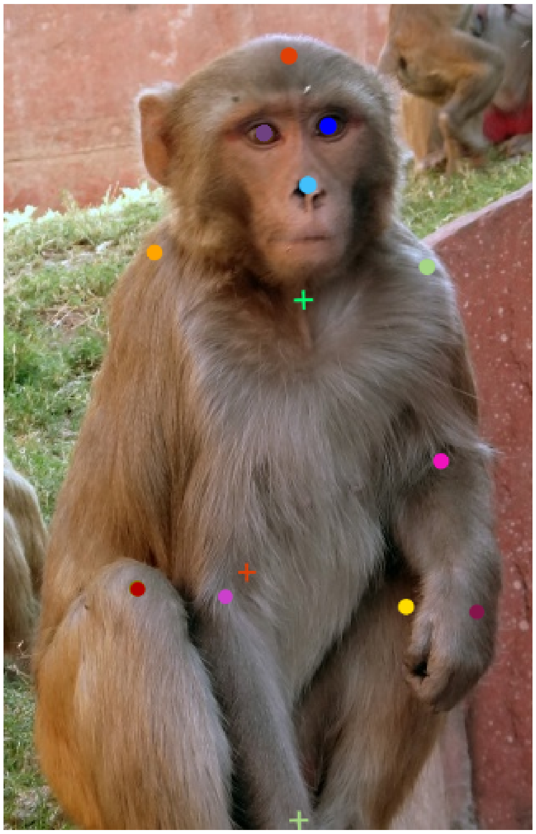
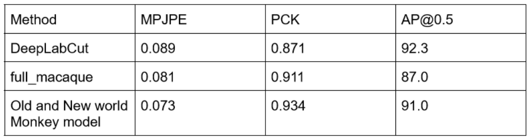


Figure 6. Joint prediction, circle indicates high confidence, X low confidence, plus is ground truth

Instead of the default 50-Layer ResNet pre-trained for object recognition tasks [7], the theDeepLabcut Model Zoo framework was utilized.The model zoo framework comprises pretrained models for specific animals and specific scenarios. The model of interest in this case was the full macaque model [10]. The full macaque model was trained using over 13,083 annotated images of macaque monkeys annotated with the help of non-researchers and refined by researchers working with macaques [8]. Although we know that the training images were largely obtained from zoos and the primate research institute of Kyoto, there is no available information regarding the actual images used for training. Since this model would be used on a filtered dataset which contains 20 other species of non-human primates it was imperative to avoid overfitting.

In order to test the performance of the model, it was tested against a model which was trained using the dataset provided in the open monkey challenge. As mentioned in the evaluation section, a large fraction of the training set was annotated images of Japanese macaque monkeys (29,242). The Japanese macaque model was trained using 20,000 annotated images of Japanese macaque monkeys [2]. The full macaque model performed better on filtered new world monkey dataset with a MPJPE of 0.082 compared to that of the Japanese macaque model (MPJPE of 0.086). Following this the full macaque model was trained using a refined old world and new world Monkey dataset to avoid overfitting. The final model performed better with a MPJPE of 0.073. The models can be found here: https://drive.google.com/drive/folders/1YkVoXYnztih7JgFb3-XOOLfy15FizJlY



Method	MPJPE	PCK	AP@0.5
DeepLabCut	0.089	0.871	92.3
full_macaque	0.081	0.911	87.0
Old and New world Monkey model	0.073	0.934	91.0

Figure 7. MPJPE, PCK, and average precision metrics for the baseline model versus the family specific models

6.2.2 Ape Model

As seen in Figure 8 across most joints, the ape model performs better than the baseline general model for all primates.

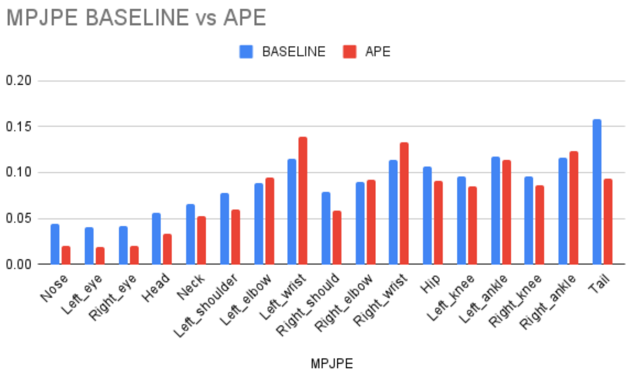


Figure 8. MPJPE Comparison between Baseline and Ape models, lower is better

In fact, our Average MPJPE (eq. 1) across all joints was 0.077, where as the baseline model was 0.089. This is in line with what we expect when we train against solely apes we shold expect a lower error in joint positioning. However as shown in Figure 8 on harder to distinguish features the advantage of our model and that of the baseline seems to fade or, in some cases, be completely lost. This is most likely due to the fact that when training a DeepLabCut model we cannot pass in the visibility parameter which was given in the training annotations, thus when our model analyzes not visible joints the result is a low confidence prediction on where that joint is on the primate. An example of this behavior is shown in Figure 9.In other metrics our model preformed just as well as in the MPJPE metric. Our Average Precision (eq. 3) was 91.1% where as the baseline 92.3%. This is likely due to the fact that our model is not too robust for handling multiple primates in a single bounding box, thus reducing our precision as it may detect different joints on different primates in the image. With a tighter bounding box we perhaps would get better results in this regard. For Probability of Correct Keypoint (eq. 2), the Ape model had an average 0.977 probability where as the baseline had a 0.871. Ape Model code can be found here: https://drive.google.com/drive/folders/1BfNd84_C2RBKrHY-6fWdhjBQBuDuMexw?usp=sharing

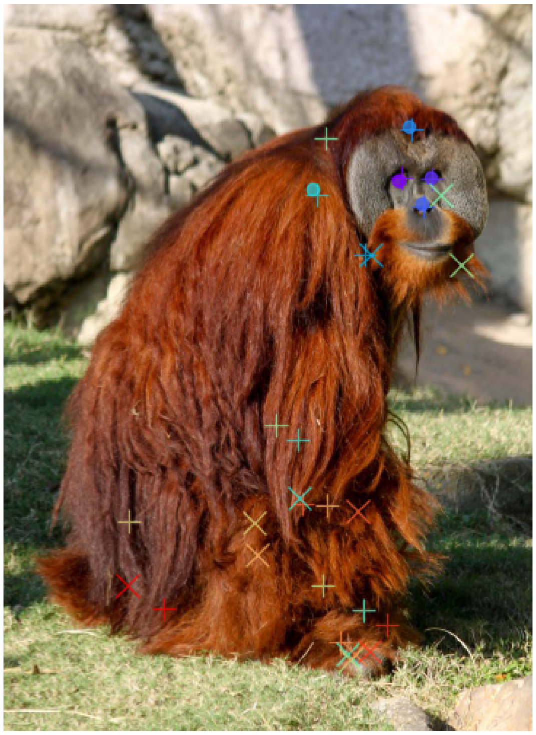


Figure 9. Joint prediction, circle indicates high confidence, X low confidence, plus is ground truth

7. Conclusion

This project was largely focused on the OpenMonkeyChallenge, using the baseline method described in section 3 and expanding upon it by creating pose estimation models specific to each primate family. The idea behind this methodology is that each family of primate have certain characteristics that they share, apparent in the data set. Therefore, separating the families should allow more specific models to be created, that differentiate between features in the species of primates belonging to their associated families better than the baseline. The goal was to see an improvement in evaluation metrics between the baseline model and the proposed family specific models. This goal was achieved, shown by example in figure 8, which demonstrates the family specific ape model was better at classifying apes that the baseline model. This was achieved in two steps: The classification and monkey family groupings, and the primate family specific model training. Classification was done using a YOLOv5 model, and feature extraction and pose estimation was done by DeepLabCut. The evaluation results overall showed a moderate increase in accuracy for the family specific models. In some cases more error was found, but this error was mostly explainable due to reasons like the features of the wrong monkey being extracted when there were multiple primates in an image, and lack of a visibility parameter in the DeepLabCut model. Fixing the bounding box errors and including visibility would only make the results found more profound, and would show more obvious positive results. Improvements on image segmentation per monkey could reduce some of the issues observed with multiple monkeys.

References

[1] Mathis A, Cury KM, Abe T, Murthy VN, Mathis MW, and Bethge M. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *CoRR*, abs/1512.03385, 2018. 5

[2] John P Capitanio and Marina E Emborg. Contributions of non-human primates to neuroscience research. *The Lancet*, 371(9618):1126–1135, 2008. 5

[3] I Wayan Agus Surya Darma, Nanik Suciati, and Daniel Siahaan. A performance comparison of balinese carving motif detection and recognition using yolov5 and mask r-cnn. In *2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 52–57, 2021. 2

[4] Sandeep Robert Datta, David J Anderson, Kristin Branson, Pietro Perona, and Andrew Leifer. Computational neuroethology: a call to action. *Neuron*, 104(1):11–24, 2019. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[8] Ned H Kalin and STEVEN E SHELTONA. Nonhuman primate models to study anxiety, emotion regulation, and psychopathology. *Annals of the New York Academy of Sciences*, 1008(1):189–200, 2003. 5

[9] Akash Kumar and Sourya Das. *Bird Species Classification Using Transfer Learning with Multistage Training*, pages 28–38. 11 2019. 2

[10] Rollyn Labuguen, Jumpei Matsumoto, Salvador Negrete, Hiroshi Nishimaru, Hisao Nishijo, Masahiko Takada, Yasuhiro Go, Ken-ichi Inoue, and Tomohiro Shibata. Macaquepose: A novel 'in the wild' macaque monkey pose dataset for markerless motion capture. *full macaque*, 2020. 5

[11] Rollyn Labuguen (P), Vishal Gaurav, Salvador Negrete Blanco, Jumpei Matsumoto, Kenichi Inoue, and Tomohiro Shibata. Monkey features location identification using convolutional neural networks. *bioRxiv*, 2018. 2, 3

[12] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, Venkatesh N. Murthy, George Lauder, Catherine Dulac, Mackenzie W. Mathis, and Alexander Mathis. Multi-animal pose estimation and tracking with deeplabcut. *bioRxiv*, 2021. 1

[13] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11, 2020. 1

[14] OpenMonkeyChallenge.com. Open monkey challenge. 1

[15] Sandeep Pandey and Saritha Sri Khetwat. A survey paper on image classification and methods of image mining. *International Journal of Computer Applications*, 169:10–12, 07 2017. 2

[16] Eko Prasetyo, Nanik Suciati, and Chastine Fatichah. A comparison of yolo and mask r-cnn for segmenting head and tail of fish. In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6, 2020. 3

[17] Thiru Siddharth, Bhupendra Singh Kirar, and Dheeraj Kumar Agrawal. Plant species classification using transfer learning by pretrained classifier vgg-19, 2022. 2

[18] Ming Tian and Zhihao Liao. Research on flower image classification method based on yolov5. *Journal of Physics: Conference Series*, 2024:012022, 09 2021. 2

[19] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019. 1

[20] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019. 2