Pose Estimation in Non-Human Primates (NHPs)

Rishi Mukherjee
University of Minnesota
mukhe100@umn.edu

Mouhamad Ali Elamine
University of Minnesota
elami018@umn.edu

Siyad Gedi
University of Minnesota
gedi0007@umn.edu

Nan Wang
University of Minnesota
wang7254@umn.edu

1. Introduction

Systems that can monitor important features in moving animals without fiducial markers, i.e., posture, have made significant strides in recent years [10] [14]. Research on the tracked species has benefited immensely from such tracking devices (e.g., rodents, flies, and fishes). Non-human primates continue to be of great interest in bio-medicine and related disciplines, such as anthropology, epidemiology, psychology, neurology, and ecology. Automated tracking can help conservation initiatives, veterinary practices, and programs for the well-being of animals [10]. Due to their uniform body texture along with exponentially large posture configurations, non-human primates position estimation is particularly difficult. Two significant advances to address the pose an estimating issue for NHPs. Existing deep learning models models are not applicable to the image samples of NHPs from the out-of-training-distribution due to their characteristics (homogeneous appearance and complex pose).

1.1. Motivation

Finding the 2 dimensional coordinates of landmarks(*keypoints in a primate's body*) and estimating the respective poses in photographs of non-human primates (NHPs) in the wild is the goal of our project. Some of the challenges faced are as follows:

- The training data is not always good/usable.
- Target may be partially in the frame of the image, therefore skewing the training set.
- Traditional feature extraction algorithms can only be so efficient.

Deep Learning, specifically Convolutional Neural Networks (CNN), have thus become the favored method for extracting features from images. With CNN's, hard coding features to find in an image is no longer required, instead the algorithm itself 'learns' from being trained with large amounts of data, continually improving the parameters [10].

2. Related Work

In a study done by Yuan Yao on the open monkey challenge, it was found that existing deep learning models such as stacked hour-glass model [6], DeeperCut [3], and AlphaPose [2] were not effective in distinguishing between monkeys due to their complex poses and homogenous appearance [13].

Classical methods such as top-down and bottom-up are major approaches for pose-estimation. Their essence is to predict the location of individuals based on object detection algorithms followed by the prediction of different body parts per cropped individual (top-down) or vice-versa. The latter approach requires two different neural networks, therefore the motivation for integrating a single network solution [11]. introduces POET model for multi-instance pose estimation that encapsulates three major components, the first being a CNN backbone for feature extraction, an encoder-decoder transformer, and a pose prediction head which outputs a vector comprising the center of mass of the detected individual with its relative body-parts while also indicating their visibility. POET accomplishes decent accuracy on the COCO dataset but still might not be optimal for primate detection (performance of 70% on Macaque dataset).

DeeperCut is another pose estimation model that can detect multiple bodies in a single image [6]. However, while both DeeperCut and this project focus on pose estimation, DeeperCut estimates human poses, not monkey poses. This model first utilizes a CNN to predict which areas of the image correspond to body parts like the head, shoulders, and knees [6]. Once all of the body parts have been predicted, then the pairwise term is calculated for every possible pair of body parts that indicates if the solution is feasible given the locations of the two body parts relative to each other, and if they belong to the same person. One of the main improvements presented by this paper is an incremental optimization strategy, which first solves the most reliable body parts, the head, and then using these solutions to find less reliable body parts like the wrists [6]. This strategy not only improved the pose estimation results of the model, but also dramatically reduced the runtime of the model as it takes less time to search for harder-to-find body parts.

RMPE is another human pose estimator, but it addresses the problem of inaccurate bounding boxes around the body. It does this by designing a SSTN, or a symmetric spatial transformer network that transforms the image of the inaccurate bounding box. Once the poses are approximated, it uses parametric pose Non-maximum suppression, which removes redundant poses based on pose similarity [3]. Before training the model, the training data is augmented using pose-guided proposal generators (PGPG), which is used to greatly augment the training data by learning the conditional of bounding box proposals for a given human pose. This method significantly outperformed the state-of-the-art methods

3. Proposed Approach

3.1. Dataset

Our dataset consists of 111529 images split into 60% training, 20% testing, and 20% validation. The images were obtained through the internet, three National Primate Research Centers, and the Minnesota Zoo. It consists of 26 species of monkeys with complex poses, and there are 17 landmarks annotated by the challenge makers. The landmarks consist of the Head, Nose, Neck, Tail, Hip, and the left/right eyes, shoulders, elbows, wrists, knees, and ankles.

Due to the inability to access the ground truth landmark annotations on the testing set, and the lack of validation options in deep lab cut, we were forced to split our training images and our validation images into smaller sets that would allow us to incorporate our proposed solution for the challenge(*more on that in the later sections*). We split the training data into directories containing 10,000 images (*we found that this worked best on google colab*), and reserved the last 6,916 images for testing our models. The validation set was left unchanged for the entirety of the project.

3.1.1 Annotation Prepping

Since all of the annotations from the image datasets were stored using the JSON we had to convert the data into a suitable format that could be accessed by our models. To do this, we wrote a python script that converted the JSON annotations into CSV file which allowed us to use the data for all of the different models. This was also useful for manual data inspection.

3.2. Baseline Method

For our baseline method we first use Uniform Manifold Approximation and Projection(UMAP) as a dimensionality reduction tool and then training a Convolutional Pose Machine(CPM) model.

UMAP is a novel method made by McInnes et al [1]. It's constructed from a theoretical framework based in Riemannian geometry and algebraic topology. With no computational restrictions on dimension embedding it's usable as a general purpose dimensionality reduction tool, and it's also competitive to t-SNE for visualization quality [1].

The CPM is the result of research combining Pose Machines with Convolutional Networks. CPM is a model where convolutional networks directly operate on the belief maps from previous stages, this produces increasingly refined estimates for part locations and it also fixes the issue of vanishing gradients during training by providing a natural learning objective function [12]. With this approach, the model outperforms other methods on standard benchmarks on datasets such as the MPII.

We arrived at this method based on a paper by Yuan Yao which used this method on the OpenMonkeyChallenge in 2021 [13]. With this approach, it is possible for us to establish a state-of-the-art, yet tested, baseline.

3.3. Proposed Method

Given that the problem at hand is to create a model that can perform well on the benchmark dataset, our solution aims to use a residual net (ResNet50) model for our architecture. We will train an artificial neural network to estimate the key points in the apes using the popular DeepLabCut algorithm [8]. DeepLabCut has been shown to be superior amongst its peers in terms of accuracy for pose estimation in animals [4]. Briefly stated, DeepLabCut is a flexible and simple algorithm that transfers the 50-layer ResNet pre-trained for the ImageNet object identification task for keypoint estimation by substituting the classification layer at the ResNet's output with the deconvolutional layers [5].

Some of the more popular use cases for pose estimation in the wild are usually tied to wildlife photography and markerless animal tracking. In their Macaque Pose dataset [7] trained a neural network to estimate keypoints in macaques. They used the DeepLabCut algorithm for markerless pose estimation in markerless animals [9] [8]. An issue they had with their neural network at the time was that the model would not work for multiple primates. It has since been adapted for multiple targets in a frame which makes it the ideal model for us to use for our neural network.

A key thing to note before moving on to the methodologies used in this experiment, there were a few challenges that we had to plan for. The first challenge was that Deep Lab Cut does not use validation in their training process. This was explicitly stated by M.W. Mathis(*leading author and researcher of DeepLabCut* [8]. The second bigger challenge was something we found out after the training process. This was the fact that DeepLabCut does not perform very well when there is multiple primate occlusion - this is when there are two primates in the image and one of the primates are occluding the other primate or vice versa.

3.3.1 Pseudo-validation

To get around DeepLabCut's 'No Validation' approach, we thought of a novel approach to splitting the training the models. We decided to train two models simultaneously: one of the models would be 'over-trained' on the validation set (*22,000 images*) and one of the models would be 'over-trained' on the training set (*50,000 images*). Once we had trained the respective models for 12 hours or 11 epochs, we saved those snapshots. Then we switched the models and fed the *validation* model to the training data, and fed the *training* model to the validation data. We then trained the two models for 4 more epochs. The two models were DLC train-Val (*this model started on the training set and finished on the validation set*) and DLV Val-train (*this model began on the validation set and finished on the training set*).

The reasoning behind this approach was to attempt to initially over-fit each of the models to their respective datasets. Then when we switched the models, we wanted the models to 'reinforce' their correctly identified features, and 'forget' their wrongly identified features.

What we found was that this helped the models perform really well on correct predictions, but our outliers were misidentified by a larger amount.

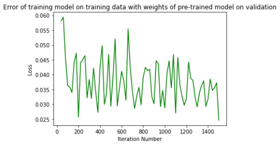
3.3.2 Multi Primate Occlusion

When the target primate is occluded by another primate, top down models (*such as the model used in this experiment*) seem to struggle to distinguish between the two primates. This is amplified when the two primates are of the same species and have similar characteristics. what we see as a result is that the landmarks are correctly identified, but they are split across multiple primates.

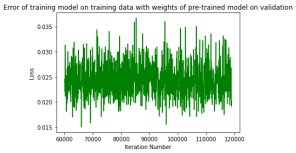
3.4. Benchmark Evaluation Protocol

The OpenMonkeyChallenge evaluates submissions with three standard metrics: mean per joint position error (MPJPE), probability of correct keypoint (PCK) metric at error tolerance, and average precision (AP) based on object keypoint similarity (OKS) [13].

MPJPE is the normalized error between the detection and ground truth, and this is calculated separately for each of the 17 landmarks. For this measure, the smaller the better.



(a) Loss for first 1500 iterations DLC train-Val model



(a) loss of DLC train-Val model

$$MPJPE_i = \frac{1}{J}\sum_{j=1}^{J}\frac{\|\hat{x_{ij}} - x_{ij}\|}{W}$$

PCK is the detection accuracy given the error tolerance. For PCK, a tolerance has to be provided that will be checked against the detection error. An example of a PCK metric would be PCK@0.2. For this measure, the bigger the better.

$$PCK@\epsilon = \frac{1}{17J}\sum_{j=1}^{J}\sum_{i=1}^{17}\delta\left(\frac{\|\hat{x_{ij}} - x_{ij}\|}{W} < \epsilon\right)$$

AP measures the detection precision, and it is also given a tolerance. It compares the given tolerance with the OKS measures, which is the keypoint similarity. OKS is different from PCK because it also takes into account the per landmark variance [13]. For AP, the bigger value the better.

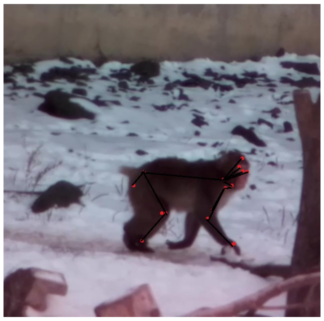
$$AP@\epsilon = \frac{1}{17J}\sum_{j=1}^{J}\sum_{i=1}^{17}\delta\left(OKS_{ij} \geq \epsilon\right)$$

$$OKS_{ij} = exp\left(-\frac{\|\hat{x_{ij}} - x_{ij}\|^2}{2W^2k_i^2}\right)$$

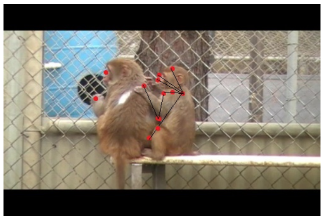
4. Results and Conclusion

4.1. Qualitative Results

The loss dies after the first 10 iterations (from 0.2 to 0.05 approximately). It is then fluctuating around 0.02 on the pre-trained models after 60k iterations. When visualizing the results of the proposed model on example images of monkeys, we are able to see that the model is able to accurately pinpoint the different landmarks across the body of the monkey like the knees, elbows, and tail. Even on the body parts that are harder to differentiate due to being in close proximity to each other, like the eyes, nose, and neck,



(a) evaluation on single primate



(a) evaluation on multiple primate

it is able to pinpoint each of these landmarks accurately. The figure on top shows an example result of an annotated image that we found that is virtually indistinguishable from the ground truth coordinates. However, some of the results were not what we expected. Based on the image above, we can see that DeepLabCut is detecting the majority of the limbs on the right monkey, but the actual monkey that's supposed to be detected was on the left.

4.2. Quantitative Results

	CPM	DLC (train-Val)	DLC (Val-train)
MPJPE	0.074	0.144	0.129
PCK@0.2	0.896	0.890	0.882
AP@e	72.9	74	70.3

The accuracy scores for our baseline and proposed methods can be seen in the table. From these results, the proposed method under-performed compared to the baseline in the MPJPE result, however we noticed outliers in the ground truth values for multiple body parts. Particularly in the case of two monkeys in the same image, it would incorrectly label the left eye to one of the monkeys, and the right eye to another assuming there is only one monkey in the image. After removing these outlier cases, we were able to achieve a result that improved the performance of the proposed method beyond the baseline with a MPJPE of 0.071 (*DLC train-Val*), and an MPJPE 0.079 (*DLC Val-train*).

4.3. Conclusion

Our proposed method performed better than the baseline method based on the given metrics. This might be because, currently, DeepLabCut is shown to be one of the best models in pose estimation. With the additional pseudo-validation performed, any potential over-fitting that could have happened was reduced and through that the results were therefore, improved. Our initial results were slightly worse than baseline because of errors in face detection when there were multiple primates and it was only after removing those outliers that we were able to see satisfactory metric values.

We were originally going to perform cross-validation rather than pseudo-validation, but DeepLabCut did not have a suitable method to incorporate cross-validation into it. Incorporating validation methods into DeepLabCut could be a way to improve this project in the future since validation methods are a powerful tool in improving the performance of a model. With validation methods, we would be able to work with smaller datasets thereby taking less storage and reduce any potential bias or over-fitting that may happen.

Another method we were considering was the introduction of the Macaque dataset into our training, with another dataset, we hope the model can be further trained, with its accuracy increasing.

The final way we are thinking to improve this project in the future is to fix the issue of DeepLabCut detecting multiple limbs from different monkeys as this was our biggest source of error. If we manage to make the model work with those kinds of images, we would be able to expand the models criterion on images and it would be much easier to train the model without having to vet out outliers beforehand.

* * *

References

[1] 2

[2] Praneet C. Bala, Benjamin R. Eisenreich, Seng Bum Michael Yoo, Benjamin Y. Hayden, Hyun Soo Park, and Jan Zimmermann. Openmonkeystudio: Automated markerless pose estimation in freely moving macaques. *bioRxiv*, 2020. 1

[3] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation, 2016. 1, 2

[4] Jacob Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair Costelloe, and Iain Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife*, 8, 10 2019. 2

[5] 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. 2

[6] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, 2016. 1, 2

[7] Rollyn Labuguen, Jumpei Matsumoto, Salvador Blanco 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. *Frontiers in Behavioral Neuroscience*, 14, 2021. 2

[8] Alexander Mathis, Pranav Mamidanna, Kevin Cury, Taiga Abe, Venkatesh Murthy, Mackenzie Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21, 09 2018. 2, 3

[9] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, 60:1–11, feb 2020. 2

[10] Stefan Schneider, Graham W. Taylor, Stefan Linquist, and Stefan C. Kremer. Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods in Ecology and Evolution*, 10(4):461–470, 2019. 1

[11] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers, 2021. 1

[12] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines, 2016. 2

[13] Yuan Yao, Praneet Bala, Abhiraj Mohan, Eliza Bliss-Moreau, Kristine Coleman, Sienna Freeman, Christopher Machado, Jessica Raper, Jan Zimmermann, Benjamin Hayden, and Hyun Park. Openmonkeychallenge: Dataset and benchmark challenges for pose estimation of non-human primates. *International Journal of Computer Vision*, pages 1–16, 10 2022. 1, 2, 3

[14] Yuan Yao, Abhiraj Mohan, Eliza Bliss-Moreau, Kristine Coleman, Sienna M. Freeman, Christopher J. Machado, Jessica Raper, Jan Zimmermann, Benjamin Y. Hayden, and Hyun Soo Park. Openmonkeychallenge: Dataset and benchmark challenges for pose tracking of non-human primates. *bioRxiv*, 2021. 1