Primate Pose Estimation with ViTPose, RandAugment, and Specie Context Tokens

Chahyon Ku, Gustav Baumgart, Maximilian Scheder-Bieschin
University of Minnesota
{ku000045,baumg260,sched088}@umn.edu

Abstract

In 2021, the Park Lab at the University of Minnesota released the benchmark challenge OpenMonkeyChallenge. This challenge aims to facilitate the creation and collection of models to automatically track non-human primate poses through various environments, and has seen 21 teams to the competition. Our team participated in this challenge with the aim of improving upon the existing models to assist researchers in fields such as biology and biomedicine, and improve their capabilities to gather insights into populations of non-human primates through pose and, by extension, gait analysis. After having reviewed existing literature on both human and non-human pose estimation, our team implemented the recently released transformer architecture, ViTPose with image augmentation and species context. Our team successfully contributed what would have been a competitive submission to the OpenMonkeyChallenge, performing 1st place on the leaderboard. Code and results can be found in our GitHub repo.

1. Introduction

OpenMonkeyChallenge is a benchmark challenge for 2D non-human primate pose estimation [27]. It consists of 111,529 photographs labeled with 17 body landmarks and is the largest non-human primate image dataset, both in number of images and number of species included. Non-human primate pose estimation is seen as more challenging than human pose estimation because non-human primates have more variation in their joint ranges and body geometry [2]. Despite this, some researchers have been able to build robust models against these challenges, and achieve comparable performances to pose estimation on primates [18] [14].

There have been efforts to reconstruct the pose of macaques in 3D, as compared to the 2D pose estimation in this challenge [2], however, the resources required to gather this data are significant and therefore limit the practicality of these techniques. Because of this limitation, being able to obtain accurate pose analysis from a single 2D image is critical for real-world applications.

Pose estimation has been applied to a wide breath of applications for humans, such as healthcare [8, 21], assisted driving [5], and video games [19]. Some of these applications require near-real time decisions to be made with small compute. Progress has been made in this area through lightweight architectures such as Fast Pose Distillation (FPD) [30]. Other applications such as AR manipulation of fine objects require more precise estimation [10]. Models created as part of the OpenMonkeyChallenge can be applied to study effects of drugs, infectious diseases, and mental illnesses on monkeys. Additionally they can be used for studies ”in the wild” such as automated monitoring of the health of wild primates [2], or understanding their social behaviors.

2. Related Work

2.1. 2D Human Pose Estimation

Human pose estimation was pioneered by Google in 2014 [23], and progress has been supported by two popular datasets: MPII [1] and COCO [15]. These datasets, the improvement of available compute, and expanding applications has lead to human pose estimation gaining increased interest and performance over the years.

Many recent work tackle human pose estimation using an end-to-end trained heatmap regression model [6]. A prime example is the convolutional pose machine which focuses on end-to-end training fully convolutional networks to classify pixels as landmark locations [24]. HRNet connected parallel high- and low-resolution convolution streams to combine the spatial and semantic information from respective streams and achieved then state of the art results in many downstream tasks including human pose estimation [22].

Transformers are an emerging architecture which has seen strong performance when applied to computer vision tasks. ViTPose, for example, achieved state-of-the-art performance by attaching a convolutional decoder head on top

of the vision transformer to directly regress the heatmap of landmark locations [26]. These transformer architectures continue to push the limits of what information can be extracted from image data.

2.2. 2D Non-human Pose Estimation

The work done on 2D non-human pose estimation is far more limited. Additionally, a substantial part of pose estimation research on primates has been done on macaque monkeys. For example, an attention-refined light-weight high-resolution network (HR-MPE) aimed at reducing the computing resources required [16].

The data set we will work with is not limited to this species. There have been some efforts towards pose estimation among multiple species as well. In one example of this, DeepLabCut has been used in order to perform pose estimation on different species of non-human primates, where they showed a slight improvement on previous work with CNNs [13].

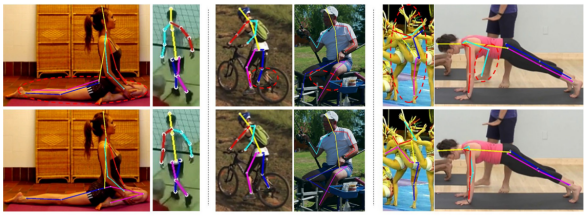
2.3. Data Augmentation

Data augmentation can lead to a significant improvement in the performance of computer vision models. One paper found that a robust augmentation and regularization scheme could have the same impact as increasing the number of samples in a dataset by a figure of magnitude for ViT [20]. This trend appears to keep up at least until 300 million samples are used for training, which is significantly more than the 111,529 samples in the OpenMonkeyChallenge.

More particularly to ViT, data augmentation techniques shown to work well were both RandAugment and Mixup [20]. Mixup linearly combines output features to create new samples and RandAugment performs a series of transformations on an image. TensorFlow's RandAugment implementation was used to measure ViT performance improvements. Additionally, the authors also tested different values for weight decay, due to the fact that AugReg could cause the models to learn over a larger number of epochs [3]. Pretrained models could also be fine tuned using AugReg later. In fact, this was more efficient and lead to better results [20].

One of the tools that was useful for training ViT, RandAugment, has a simple implementation and can greatly decrease the search space for automated augmentation. It merely takes two parameters N, the number of transformations, and M, which specifies the particular bin that contains the magnitude of each transformation (aside from the image and number of bins) and generates a sequence of transformations to perform on the image. The pseudocode that we followed is included in section 6.2. Although performance in object detection and object classification has been shown to increase using RandAugment (and also further improve as the number of random transformations grow) [7], we are not aware of this technique being tested on pose estimation.

We also considered a data augmentation technique for human pose estimation (ASDA). This particular form of data augmentation targeted poses that were especially difficult to predict due to symmetric appearance, heavy occlusion, or nearby persons, which are the current weaknesses of CNNs [4]. However, the ASDA scheme involves categorizing parts of the human body, randomly selecting them from a pool, and properly pasting them together in order to perform data augmentation [4]. Given the information we are provided in the OpenMonkeyChallenge dataset, we decided it would be too complicated to implement this data augmentation scheme.



(a) Symmetric Appearance (b) Occlusion (c) Nearby Person

Figure 1. Difficult situations for deep CNNs to perform pose estimation [4].

3. Dataset

The OpenMonkeyChallenge has 111,529 total RGB images with 17 possible landmarks per sample. The dataset builds on the scope of OpenMonkeyPose [2] and MacaquePose [14] datasets. OpenMonkeyPose has 195,228 images with 13 annotated landmarks, but is specifically designed for understanding 3D movement and only contains rhesus macaques. Additionally it was compiled using a single, controlled environment, potentially making it difficult to apply to broad applications. The MacaquePose dataset consists of only 13,083 images with 17 annotated landmarks, sourced from an number of different environments but likewise only for macaques [27].

The OpenMonkeyChallenge dataset is divided into 60/20/20 train/validation/test splits, with 66,917, 22,306, and 22,306 images respectively [27]. Each image is cropped to contain one or more monkeys, with each crop having a resolution of at least 500x500 pixels. These landmarks consist of the nose, left and right eyes, head, neck, left and right shoulder, left and right elbow, left and right wrist, hip, left and right knee, left and right ankle, and tail. Notably, this dataset is comprised of a wide variety of primate species (26 in total), which the authors categorize as New World (6) and Old World (14) monkeys, as well as apes (6). Their environments are also varied, with sources including Flickr, YouTube, three National Primate Research Centers, and the

Minnesota Zoo among others. Because of this, the OpenMonkeyChallenge provides the strongest publicly available dataset for the study of non-human primate images across domains.

Previous datasets have limitations with regards to species diversity, environments, complexity of image-capture, and dataset size, which hinders generalization of primate pose estimation. OpenMonkeyChallenge is designed to increase the ability to generalize pose estimation performance across primates, regardless of their environment [11, 29].

4. Evaluation Metrics

We will use probability of correct keypoint (PCK@ϵ with $\epsilon = 0.2$) and average mean per joint position error (MPJPE$_i$) as performance metrics. PCK@ϵ is a way of measuring how likely it is for the model to predict any joint's position. MPJPE$_i$ calculates the mean distance of the predicted joint position from the actual joint position. Both of these metrics do normalize by the size of the bounding box W.

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

$$\text{MPJPE}_i = \frac{1}{J}\sum_{j=1}^{J}\frac{||\hat{x}_{ij} - x_{ij}||}{W} \quad (2)$$

These are the metrics proposed by OpenMonkeyChallenge to evaluate model performance.

5. Baseline Method

For the baseline we implemented HRNet. This model was chosen as it is one of the best baseline models from the original OpenMonkeyChallenge paper, with average precision of 0.78. Its implementation allowed us to confirm that our pipeline works properly. The code and models used in the HRNet paper are available through their Github repository and have been broadly implemented, providing additional troubleshooting resources if needed.

HRNet was initially proposed for human pose estimation in 2019 [22]. Unlike previous methods that traversed low-to-high or high-to-low resolutions, [12], HRNet proposed a new architecture that maintained high-resolution representations through the entire model. This led to two benefits as compared to existing models (1) improved spatial precision due to not needing to recover resolution (2) improved capabilities for pose estimation due to repeated multiscale fusions [22]. Figure 3 shows that HRNet performs well against a number of state-of-the-art models.

The HRNet model architecture and weights from training on ImageNet have been implemented using PyTorch and TensorFlow allowing for easy transfer into our own model.

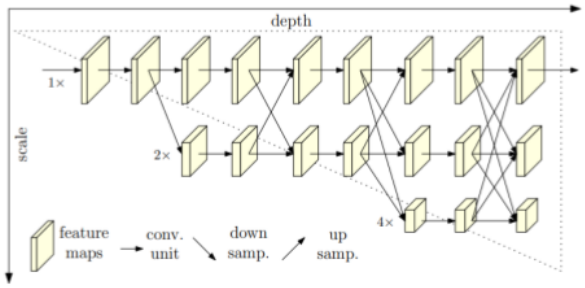


Figure 2. HRNet visual from Deep High-Resolution Representation Learning for Human Pose Estimation. [22]

Entry	Additional training Data	mAP	MOTA
ML-LAB	COCO+MPII-Pose	70.3	41.8
SOPT-PT	COCO+MPII-Pose	58.2	42.0
BUTD2	COCO	59.2	50.6
MVIG	COCO+MPII-Pose	63.2	50.7
PoseFlow	COCO+MPII-Pose	63.0	51.0
ProTracker	COCO	59.6	51.8
HMPT	COCO+MPII-Pose	63.7	51.9
JointFlow	COCO	63.6	53.1
STAF	COCO+MPII-Pose	70.3	53.8
MIPAL	COCO	68.8	54.5
FlowTrack	COCO	74.6	57.8
HRNet-W48	COCO	**74.9**	**57.9**

Figure 3. Performance of HRNet against other popular pose identification architectures. [22]

To implement our model we modified the original HRNet by adding a 17 channel regression head. These channels provide as output a heatmap of for each of the 17 landmarks used by OpenMonkeyChallenge. After transferring in the weights of the original HRNet Model, we fine-tune on the OpenMonkeyChallenge dataset, updating the weights for our specific task. To keep the initial complexity low, we started with the shallowest variant of HRNet before training deeper models.

6. Proposed Method

6.1. ViTPose

We implemented the ViTPose model [26]. This model was selected as it has not yet been applied to the task of non-human primate pose estimation and has recently reached state-of-the-art performance for human pose estimation. Our goal with this model was to improve upon the current best performing model submitted to OpenMonkeyChallenge. The paper was submitted to Arxiv in April 2022, to document the performance of applying the excellent performance seen from transformers in visual recognition tasks to pose estimation. As their paper shows, they achieved a significant performance improvement against other state-of-the-art methods on the COCO validation set.

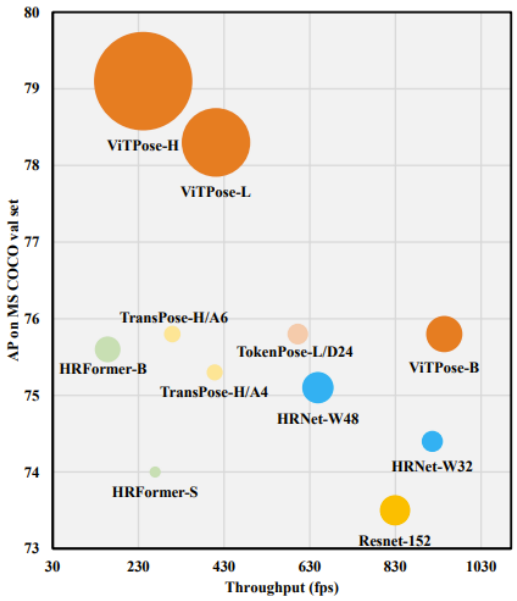


Figure 4. Comparison of ViTPose against other state-of-the-art methods using the MS COCO validation set. The size of each bubble represents the number of model parameters, the horizontal axis throughput, and the vertical axis precision. [26]

The ViTPose architecture consists of non-hierarchical vision transformers as backbones which serve to extract feature maps for the given instances of people. The backbones they employed were trained on masked image modeling pretext such as MAE to provide a strong weight initialization. Once the feature maps are extracted they are passed into a lightweight decoder which processes them by upsampling the feature maps and regressing the associated heatmaps with respect to the landmarks, similar to the methodology for ResNet simple pose baseline [25]. This decoder consists of just two deconvolution layers and one prediction layer. An overview of this architecture is shown in Figure 5.

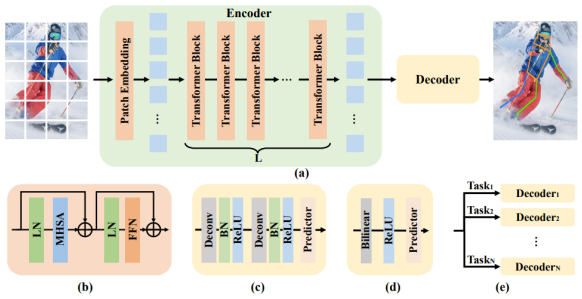


Figure 5. Visuals from ViTPose (a) The ViTPose model. (b) The transformer block. (c) The classic decoder. (d) The simple decoder. (e) The decoders for multiple datasets. [26]

Outside of its performance the ViTPose shows that it is adjustable in regards to its simplicity, scalability, flexibility, and transferability. These traits should help us be able to cater our implementation better to our application and available compute. In the discussion section of their paper Xu et. al. share their belief in the success of ViTPose when applies to animal pose estimation; one of the reasons our group decided to pursue this as our proposed method.

6.2. Randaugment

We used RandAugment [7] ($M = 0, 2, 4$, $N = 3$, and $num\_bins = 10$) along with ViT. Augmentation for the images was done in the same way as described in the pseudocode below, but we did have to perform operations differently for the output heatmap generated for each landmark location. This is because some operations involved transformations on the coordinates of the image and others (such as brightness and contrast) instead performed operations using filters. We applied the affine transforms to the output heatmaps only when the former kind of transformation was used.

Algorithm 1 RandAugment

Randaugment(N,M, num_bins, img)

$bins \leftarrow [0 \cdot \frac{0.99}{num\_bins}, 1 \cdot \frac{0.99}{num\_bins}, \ldots, num\_bins \cdot \frac{0.99}{num\_bins}]$
$magnitude \leftarrow bins[M]$
for $i = 1 \ldots N$ **do**
    $t \leftarrow magnitude\cdot$ randomChoice([Identity, ShearX, ShearY, TranslateX, TranslateY, Rotate, Brightness, Color, Contrast, Sharpness, Posterize, Solarize, AutoContrast, Equalize])
    if t is signed **then**
        $t \leftarrow t\cdot$ randomChoice($[-1, 1]$)
    end if
    $img \leftarrow$ apply_operation(t, img)
end for
return img

In an attempt to keep the actual location of the joints from moving out of the bounding box (given that rotation is one of the transformations), we used a smaller magnitude for all operations. The rotations, which were capable of driving the points of interest out of the bounding box, were minor. We could visually confirm that (on a subset of the augmented data), the augmentation did not lead to the joints falling out of the bounding box, so we consider this possibility unlikely.

6.3. Specie Context Tokens

The motivation behind specie context tokens is to utilize the provided specie information of the monkeys. A vanilla ViT prepends a "class" embedding before the image

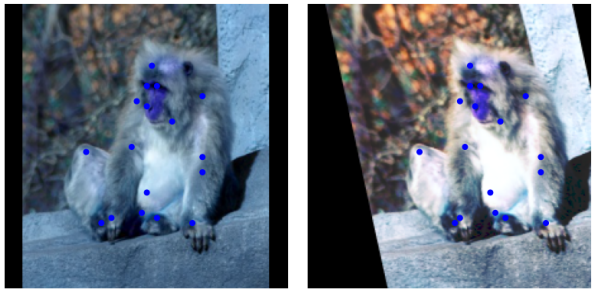


Figure 6. Original sample on left; RandAugmented sample on right.

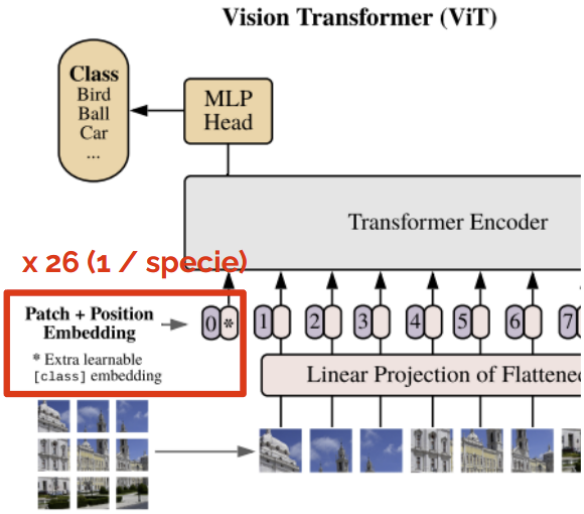


Figure 7. Modified diagram from ViT [9] showing the idea of specie context tokens

tokens to both extract image class information from image patches and allow the image patches to attend to the task context of image classification. We hypothesize that by having 26 different learnable "class" embeddings, 1 per specie, we allow the ViT to learn the relevant context information for each specie and improve performance. This method adds a negligible number of parameters (26K) and does not add any computation, while still encouraging the model to learn specie-specific knowledge of joint apperances and locations.

7. Experiments and Results

We conducted experiments by fine-tuning ImageNet-pretrained models from the timm library. For all our experiments, we used images resized to 224 x 224 and batch size of 16 to train on a single A5500 GPU using the AdamW optimizer with learning rate 1e-4. As the competition website stopped accepting submissions, and we had no access to the test data, we compared the performance of each model

#	User	Entries	Date of Last Entry	MPJPE ▲	PCK@0.2 ▲	PCK@0.5 ▲	mAP ▲
1	TiSpH	8	04/29/22	0.047 (15)	0.964 (1)	0.996 (1)	0.925 (1)
2	csci5561shihaoyi	2	04/26/22	0.053 (14)	0.957 (2)	0.994 (2)	0.901 (2)
3	mahaj068	1	04/28/22	0.071 (12)	0.939 (3)	0.991 (3)	0.786 (5)
4	chan1975	16	04/28/22	0.068 (13)	0.920 (4)	0.983 (5)	0.838 (3)
5	openmonkey	5	12/08/21	0.075 (11)	0.918 (5)	0.987 (4)	0.789 (4)
6	choix709	3	04/28/22	0.105 (8)	0.872 (6)	0.978 (7)	0.590 (8)
7	openmonkeychallenge	4	12/10/21	0.101 (9)	0.866 (7)	0.976 (8)	0.657 (7)
8	baseline_sb	1	08/13/21	0.095 (10)	0.842 (8)	0.980 (6)	0.668 (6)
9	spani084	1	04/29/22	0.199 (7)	0.711 (9)	0.887 (10)	0.449 (9)
10	Avinash_Akella	2	04/28/22	0.228 (4)	0.676 (10)	0.895 (9)	0.411 (10)
11	grazz018	8	04/29/22	0.219 (5)	0.665 (11)	0.871 (12)	0.380 (11)
12	tarn1108	3	04/28/22	0.213 (6)	0.596 (12)	0.872 (11)	0.260 (12)
13	Yes	5	04/30/22	0.725 (3)	0.014 (13)	0.199 (13)	0.000 (13)
14	openmonkey_unet	3	12/12/21	1.001 (2)	0.010 (14)	0.157 (14)	0.000 (13)
15	steven.moore128	3	12/11/21	1.286 (1)	0.000 (15)	0.025 (15)	0.000 (13)

Figure 8. OpenMonkeyChallenge Leaderboard from 12/18/2022

by evaluating on the validation data with the top performing models on the current leader-board (latest submission April 2022). Since the main metric for the competition is MPJPE, we will be focusing on this to evaluate our performance.

7.1. HRNet Baseline

Models	MPJPE	PCK	AP	# Params
HRNet w18	0.065	0.927	0.805	**10M**
HRNet w32	0.063	0.933	0.814	30M
HRNet w48	0.062	0.936	0.817	67M
HRNet w64	**0.061**	**0.938**	**0.820**	118M

Table 1. HRNet Validation Results

For our baseline, we fine-tuned 4 versions of HRNet as provided by the timm library: w18, w32, w48, and w64. We trained them for 20 epochs and evaluated the model that had the lowest validation loss.

This brings us to 3rd place in the leaderboard at 0.061 MPJPE for HRNet w64.

7.2. ViTPose

Models	MPJPE	PCK	AP	# Params
s8u2	0.055	0.957	0.833	105M
s4u2	0.058	0.949	0.819	105M
s16u2	0.058	0.958	0.813	105M
s8u3	0.051	0.959	0.852	115M
s8u4	**0.051**	**0.959**	**0.855**	124M

Table 2. ViTPose Validation Results

We fine-tune ImageNet-pretrained ViT-base for our ViTPose unless otherwise indicated. As our first set of experiments with ViTPose, we experiment with two hyperparameters: heatmap size and number of UpConv (transpose convolution) layers. To speed up the process of finding the right

parameter, we do not train until convergence but rather train for 5 epochs.

First, we observed that ViTPose is much slower to train and more sensitive to the size of the target heatmap. Hence, to speed up the training process, we experiment with different heatmap sizes: 4, 8, and 16. This size **s** defines the standard deviation of the 2D gaussian kernel centered at the groundtruth joint locations. By sweeping multiples of 2 from 1 to 16, we report that 8 was the best size to use with MPJPE of 0.055.

Second, we observed that the output of ViTPose is half as big as that of HRNet. This is because the feature size of ViTPose is 1/16 of the input image size, hence having 2 UpConv layers [28] that upscales the size by a factor of 2 (as the original ViTPose paper suggests) regresses a heatmap that is 1/4 of the input image. We hypothesize that this loss of spatial resolution may result in poor performance and verify that simply increasing the number of UpConv layers, **u**, will restore this spatial resolution and improve results.

This brings us to 2nd place in the leaderboard at 0.051 MPJPE for ViTPose with 4 UpConv layers.

7.3. ViTPose with RandAugment and Specie Context Tokens

Models	MPJPE	PCK	AP	# Params
s8m0u2	0.055	0.958	0.833	105M
s8m0u2c	0.054	0.959	0.833	105M
s8m2u2	0.052	0.963	0.844	105M
s8m2u2c	**0.050**	**0.964**	**0.850**	105M
s8m4u2	0.052	0.962	0.843	105M
s8m4u2c	0.052	0.963	0.844	105M

Table 3. ViTPose with RandAugment and Specie Context Tokens Validation Results

To further improve performance, we propose two additions to ViTPose: RandAugment and Specie Context Tokens. All models in this section are trained for 10 epochs.

First, we observed that the validation performance plateaus and starts decreasing around the 5th epoch. Adding RandAugment slows down the training, but allows the model to learn past the non-augmented limit where the model starts to overfit. We vary the magnitude **m** of the augmentation, which defines how skewed or jittered the color and shape of the image is, to 0 (no augmentation), 2, 4 and find that magnitude of 2 is the most effective in improving the performance.

Second, we propose specie context tokens to utilize the provided specie information. Adding context **c**, even though it requires a minuscule (26K) number of extra parameters and no additional computation, improved the performance for all magnitudes of data augmentation.

7.4. Final Results with Large Models

Models	MPJPE	PCK	AP	# Params
base-u3	0.0468	0.9677	0.8699	**115M**
base-u4	0.0463	0.9671	0.8737	124M
large-u3	0.0458	0.9690	0.8741	355M
large-u4	**0.0449**	**0.9690**	**0.8793**	371M

Table 4. Large Models Validation Results

Using the methods and parameters we verified in the previous two sections, we train larger models for 20 epochs to report our final performance. To specify, heatmap size **s** of 8, rand augment magnitude **m** of 2, and specie context prefix token are all used for ViT-base and ViT-large with 3 and 4 UpConv layers.

This brings us to 1st place in the leaderboard at 0.045 MPJPE for ViTPose-large with 4 UpConv layers.

8. Conclusion

In this paper we successfully implemented a custom ViTPose model, a state of the art Vision Transformer, to the challenge of pose estimation of a wide range of primates. We supplemented the use of the largest public collection of primate images through RandAugment, which has not been done in the context of pose estimation. Also, we prepend a species context token to let the model learn a context embedding for each specie. Through this work we were able to achieve first-place results as compared to the published competition results.

Looking forward, future work on this project would explore replacing the vision transformer with more complicated variants, such as the Swin Transformer [17], additionally connecting to additional university resources seems intuitive. A more complicated future step is implementing the ASDA augmentation approach mentioned in the data augmentation section. This is expected to significantly improve performance, however requires manual annotation work.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[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, 2

[3] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and

scaling strategies. *Advances in Neural Information Processing Systems*, 34:22614–22627, 2021. 2

[4] Yanrui Bin, Xuan Cao, Xinya Chen, Yanhao Ge, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Changxin Gao, and Nong Sang. Adversarial semantic data augmentation for human pose estimation. In *European conference on computer vision*, pages 606–622. Springer, 2020. 2

[5] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4661–4670, 2017. 1

[6] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. *CoRR*, abs/1609.01743, 2016. 1

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 2, 4

[8] Vandad Davoodnia, Saeed Ghorbani, and Ali Etemad. In-bed pressure-based pose estimation using image space representation learning. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Jun 2021. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 5

[10] Niklas Gard, Anna Hilsmann, and Peter Eisert. Combining local and global pose estimation for precise tracking of similar objects. 2022. 1

[11] Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. Automated pose estimation in primates. *American journal of primatology*, page e23348, 2021. 3

[12] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, 2016. 3

[13] Rollyn Labuguen, Dean Karlo Bardeloza, Salvador Blanco Negrete, Jumpei Matsumoto, Kenichi Inoue, and Tomohiro Shibata. Primate markerless pose estimation and movement analysis using deeplabcut. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 297–300. IEEE, 2019. 2

[14] 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*, 14:581154, 2021. 1, 2

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1

[16] Sicong Liu, Qingcheng Fan, Shanghao Liu, Shuqin Li, and Chunjiang Zhao. An attention-refined light-weight high-resolution network for macaque monkey pose estimation. *Information*, 13(8):356, 2022. 2

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021. 6

[18] Salvador Blanco Negrete, Rollyn Labuguen, Jumpei Matsumoto, Yasuhiro Go, Ken-ichi Inoue, and Tomohiro Shibata. Multiple monkey pose estimation using openpose. *bioRxiv*, 2021. 1

[19] Manisha Patel and Nilesh Kalani. A survey on pose estimation using deep convolutional neural networks. *IOP Conference Series: Materials Science and Engineering*, 1042(1):012008, Jan 2021. 1

[20] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 2

[21] Jan Stenum, Kendra M. Cherry-Allen, Connor O. Pyles, Rachel D. Reetzke, Michael F. Vignos, and Ryan T. Roemmich. Applications of pose estimation in human health and performance across the lifespan. *Sensors*, 21(21), 2021. 1

[22] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1, 3

[23] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. 2013. 1

[24] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016. 1

[25] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *CoRR*, abs/1804.06208, 2018. 4

[26] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022. 2, 3, 4

[27] 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, 2

[28] Chengxi Ye, Matthew Evanusa, Hua He, Anton Mitrokhin, Tom Goldstein, James A. Yorke, Cornelia Fermüller, and Yiannis Aloimonos. Network deconvolution. *CoRR*, abs/1905.11926, 2019. 6

[29] Haozheng Yu. Segmentation and dense keypoints estimation of monkeys. 2021. 3

[30] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1