CV for Primates

Rundong Luo[1], **Yiran Geng**[1], **Zebin Yang**[1]
[1]School of EECS, Peking University
{rundong_luo, gyr, 2000012701}@stu.pku.edu.cn

Abstract

Understanding the behavior of primates is a long-standing and critical task for multiple disciplines. Although recent advances in deep learning provide a new paradigm for handling this task, most existing methods only leverage simple architecture such as CNN. In this report, we address three representative tasks on primates: detection, identification, and pose estimation. We explore multiple architectures and build strong baselines. Based on these baselines, we introduce novel approaches and achieve state-of-the-art results. Extensive experiments demonstrate the superiority of our methods. All our code and results are available at `https://github.com/Red-Fairy/CV-project-22Fall`.

1 Introduction

Understanding the behavior of primates is vital for primatology, psychology, and biology since primates are important model organisms. The challenge of accurately measuring animal behavior has been a longstanding issue, primarily due to the time and effort required for manual observation and the difficulties of long-term monitoring in natural environments. Recent advancements in technology, such as computer vision, machine learning, and robotics, coupled with breakthroughs in deep learning, have made it possible to track the behavior of primates through the use of CNN-based software that can identify objects in images.

The capability of automatically monitoring and analyzing primate behavior will significantly assist researchers in cognitive science. To achieve this, when given a video of wild chimpanzees that have some annotations, the first step is to identify all the individuals by their location, identity, and activity through detection, identification, and 2D pose estimation. After that, it will be possible to create a quantitative representation of the social network and infer the social connections among the primates.

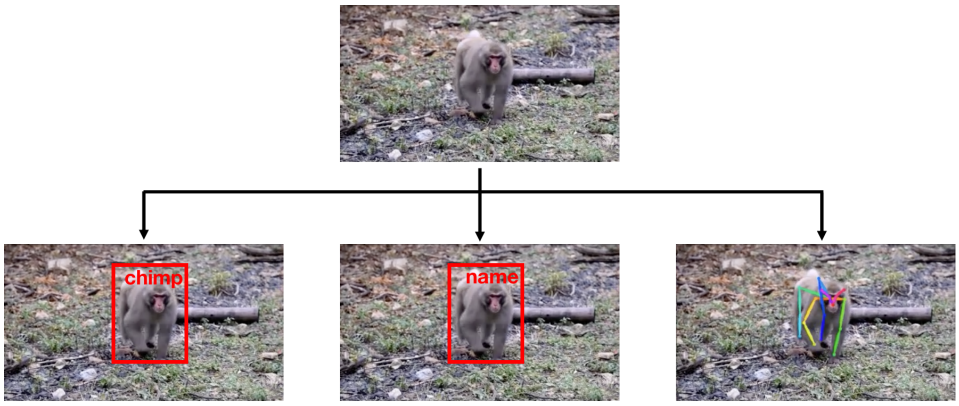
Therefore, we address these representative tasks on primates in this report. An illustration to these tasks is shown in 1. Unlike existing methods [30] only leverage simple architecture such as CNN, we explore more advanced techniques, including YOLO [16], contrastive learning [3], MacaquePose [21], etc. We build solid baselines upon these novel approaches. Afterwards, we present our improvenets to these baselines and demonstrate the effectiveness of our methods through extensive experiments.

The remainder of this report will be organized as follows: Sec. 2 presents the related works of the three tasks involved. Sec. 3 to Sec. 5 comprehensively describes the baselines and our method for detection, identification, and pose estimation. Experimental results and analysis of the results will be given in Sec. 6. Finally, Sec. 7 presents the conclusions and discussion of future works.

2 Related Works

2.1 Object Detection

The task of object detection is to find all interested objects in the image and determine their categories and positions, which is one of the core issues in computer vision. R-CNN [29] is the first algorithm that successfully applies deep learning to detection, extracting features from candidate regions for judgment using a CNN architecture. Fast R-CNN [10] optimizes the repeated computation when



(a) detection (b) identification (c) 2D pose estimation

Figure 1: An illustration to the tasks involved in this report.

R-CNN performs feature extraction for all regions, improving the performance. Faster R-CNN [31] adds a neural network to find interested areas, further reducing the time consumption of the model. YOLO [16] uses the idea of regression, uses the whole picture as the network's input, and directly returns the bounding boxes and the categories of the objects on multiple positions of the image.

2.2 Identification

Long-tail Classifcation. Training samples typically exhibit a long-tailed inter-class distribution in real-world applications, including primate identification, where a small portion of classes accounts for the majority of sample points. Such class imbalance of training samples could make training vanilla-supervised deep networks challenging. Existing methods of long-tailed learning can be classified into class re-balancing, data augmentation, and representation learning. Class re-balancing methods [23, 18] utilize re-sampling techniques to guarantee a balanced sample size during training. Logit adjustment operations [22] are also considered. Data augmentation methods [42, 6, 26] aim to enhance the size and quality of the dataset. Representation learning methods [8, 14, 40, 43, 17] aim at designing loss metrics (*e.g.,* contrastive loss [43, 17]) for establishing similarity or dissimilarity between classes. Interesed readers may refer to [41] for comprehensive surveys.

As for the primate identification task, both the training and testing sets follow the long-tailed distribution. Data scarcity is also a problem. Therefore, simple re-sampling methods may not work well as they may bring discrepancies between the train and test distributions. Based on the characteristics of the task, we propose a method compound method that enjoys the merits of three kinds of methods, thus achieving satisfactory results.

Contrastive Learning. Contrastive learning [3, 13] is a popular self-supervised learning paradigm that contrasts positive and negative image pairs. Given a mini-batch $\mathcal{X}$, we draw two positive samples x, x^+ for each $\bar{x} \in \mathcal{X}$ following the augmentation distribution $A(\cdot|\bar{x})$, and draw M independent negative samples $\{x_m^-\}_{m=1}^M$ from the marginal distribution $A(\cdot) = \mathbb{E}_{\bar{x}} A(\cdot|\bar{x})$. Then, we train an encoder $g : \mathcal{X} \to \mathcal{Z}$ by the widely adopted InfoNCE loss [25] using the augmented data pair $(x, x^+, \{x_m^-\})$

$$\mathcal{L}_{\text{NCE}}(x, x^+, \{x_m^-\}; g) = -\log \frac{\exp(\text{sim}(g(x), g(x^+))/\tau)}{\sum_m \exp(\text{sim}(g(x), g(x_m^-))/\tau)}, \tag{1}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity between two vectors, and τ is a temperature hyperparameter. Although designed for self-supervised settings, recent literature [43, 19] has extended contrastive learning to supervised scenarios when labels are available.

2.3 Pose Estimation

Pose estimation problems can be divided into two main categories: 2D pose estimation and 3D pose estimation. As the name implies, the former predicts a 2D coordinate for each key point; the latter predicts a 3D coordinate for each key point, adding one-dimensional depth information. For 2D pose

estimation, most current research is on multi Person pose estimation, i.e., each image may contain more than one person. There are usually two types of ideas to solve the problem: top-down and bottom-up.

Top-down Approaches. In conjunction with recent classic work [27, 9, 12, 5, 38, 32], the idea of top-down is to first perform target detection on the image to find all the people (obtain the bounding box for each people); then crop the people out of the original image, resize them and input them to the network for pose estimation. In other words, top-down transforms the problem of multi-person pose estimation into a problem of multiple single-person pose estimation. Mask R-CNN[12] adds a pose estimation module for end-to-end training. Fang *et al.* [9] optimizes the problem of inaccurate bounding box search using transformer architecture. PandaNet [1] uses the anchor-based method to improve the 3d pose estimation method. Chen *et al.* [5] uses different networks to process different information. Specifically, they use two networks to process coarse-grained and fine-grained information separately and eventually integrate them, ensuring the information's integrity and allowing for increased accuracy and generalization. Sun *et al.* [32]'s approach is multi-stage, and they mainly use high resolution networks for pose estimation optimization. The primary consideration is preventing information loss during high and low resolution changes. The constant fusion allows for better retention of detailed information. Bertasius *et al.* [2] propose to not only learn in images but also about changes and relationships in time series from videos to acquire better pose estimation skills.

Bottom-up Approaches. Bottom-Up Approaches, *i.e.,* part-based frameworks, first detect each key part of the human body and then stitch the detected parts together to form a human figure. The disadvantage is that different parts of different people are stitched together as one person. Representative methods [4, 11, 35, 28, 15, 20, 33] often sacrifice accuracy but increase processing speed. Another feature of these methods is that they are very sensitive to the size of the human sample in the picture, and as there is no function to reset the size of the individual, it often introduces a high level of error when the proportion of the human body is very small and even fails to recognize the human pose.

One-stage Approaches. In addition to top-down and bottom-up approaches, there are other methods [39, 24, 34, 37] that are not two-stage. Specifically, instead of first detecting the body's key points, they locate the body and detect the key points simultaneously, which can increase efficiency. However, because of the complexity of the representation, their accuracy is similar to that of the first two approaches.

Pose Estimation for Primate Since the application scenario for pose estimation in animals is different from human pose estimation, there needs to be more mature and complete work on primates. In recent years, as wildlife conservation and monitoring have become more sophisticated, there are many artificial intelligence algorithms dedicated to wildlife. Among them, MacaquePose [21] annotates all kinds of primates' poses. Based on this, mmpose [7] also provides some corresponding training models.

3 Detection

The goal of detection is to train a model to judge whether there are chimpanzees in the input image by giving confidence scores and bounding boxes indicating the specific locations of the chimpanzees.

3.1 Data Collection

A total of 1351 images and corresponding bounding boxes are given in the whole data set, including 1251 images as the training set and 100 as the test set.

3.2 Data Cleaning

In the whole dataset, there are no chimpanzees in some images. The label textfiles corresponding to these images are empty, and these data cannot participate in the training process. We scan all the label text files, delete the empty text files, and delete the images with the same name to get a clean data set. After cleaning the data, 1188 images remain in the training set, and 94 images remain in the test set.

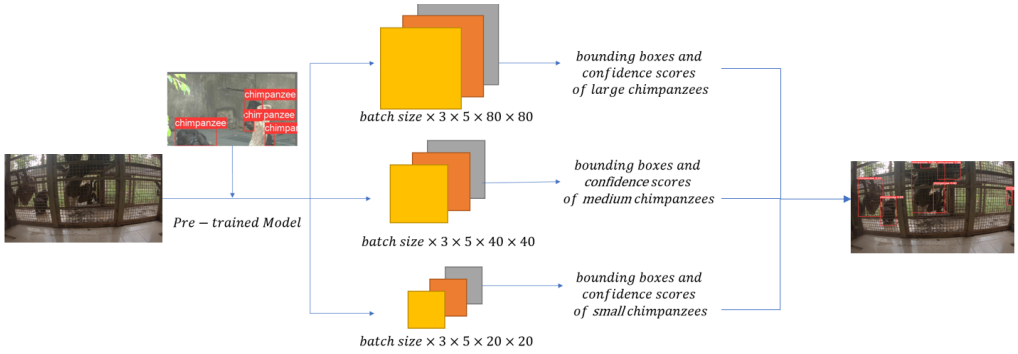


Figure 2: Pipeline of primates detection.

3.3 Method

To solve this detection problem, we used the open-source yolov5s model for training, using SGD as the optimizer in the training process. Input the images into the model, and we can get the bonding boxes and confidence scores of different chimpanzees from different outputs of the model. The pipeline of our detection module is shown in Figure 2. This section will introduce some of our optimization points based on the original model.

Number of Training epochs. Considering that the data in the training set is not so large and the model's initial number of training epochs is as significant as 300, the model may be overfitting. Therefore, after the end of each training epoch, we calculate the mAP of the current model on the test set. After that, for each epoch, we calculate the average value of mAP of five models on the test set obtained from this epoch and the past two and the last two epochs. When the training reaches 90 epochs, we can get the maximum average value of mAP, demonstrating the conjecture of model over-fitting. So we changed the number of training epochs to 90, which improved the model's performance. Results are shown in Sec. 6.1.

Size of the model. The network we originally trained uses the yolov5s architecture. The model's size is small, and the model's ability to extract features from the input is limited. So we used yolov5m and yolov5l for training, respectively. For each model, the optimization method of the number of training epochs is the same as yolov5s.

4 Identification

The goal of identification is to train a neural network that can predict the correct identity of a primate given its image.

4.1 Data Collection

The training set for the identification task includes 662 images; each corresponds to a bounding box given in the original training set. The test set contains 275 images and is gathered similarly. The dataset contains 17 different individuals, i.e., a 17-class classification problem.

4.2 Method

The main difficulties of this task lie in the long-tailed distribution and label scarcity. Specifically, a chimp named "Azibo" accounts for about 40% of all samples in the training set (Figure 3). In this section, we introduce our novel framework to address such challenges.

Data Augmentation. Data augmentation is a widely applied technique to enhance the dataset's variety. However, fierce data augmentation will even bring performance degradation. There are several reasons for this phenomenon. The first reason is the tightness of bounding box annotations, which would make random cropping produce worse samples. The second reason is the train-test distribution gap. Since the train-test distribution is almost identical, adding fierce augmentations would enlarge such a gap and distort the semantics, thus harming the overall performance. We ablate the effect of different data augmentations in Table 1 and show that only horizontal flip (default) benefits the overall

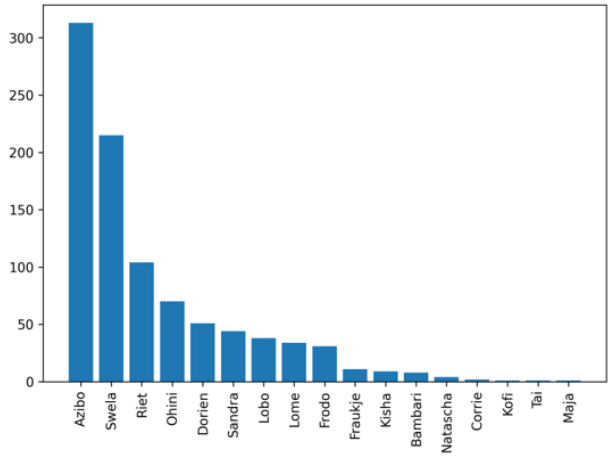


Figure 3: Illustration on the training sample distribution.

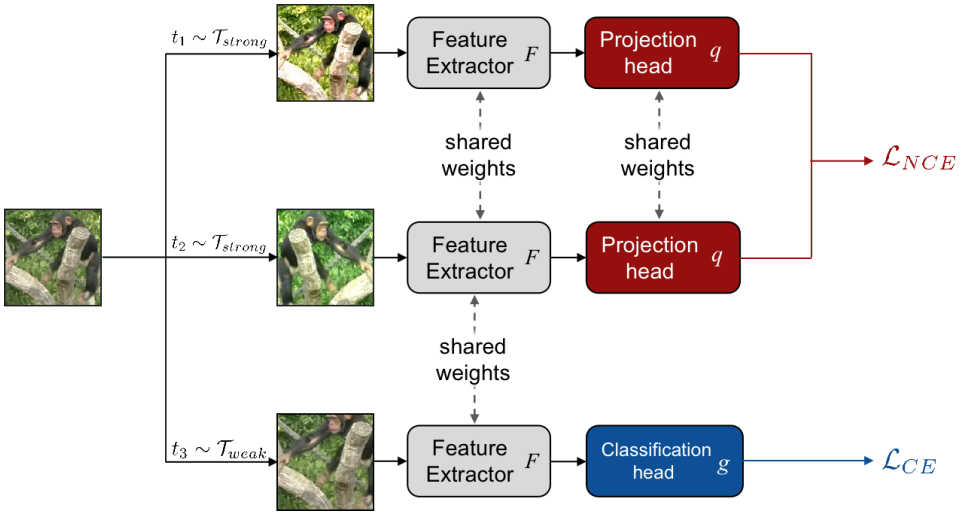


Figure 4: Pipeline of the classification module.

performance under vanilla supervised learning settings. To solve such a problem, we propose a novel strong-weak augmentation scheme where augmentation is drawn from two distinct augmentation distributions.

Table 1: Ablation on the effect of each data augmentation. Default augmentation (horizontal flip) is used in all experiments. Weak crop refers to resizing (256×256) and cropping (224×224); fierce crop refers to "torchvision.transforms.RandomResizedCrop$(224, (0.5, 1.0))$"; color jitter uses parameter $(0.2, 0.2, 0.2, 0.05)$; the probability of applying grayscale is 0.2; the range of rotation is $(-15°, 15°)$.

Augmentation	default	weak crop	fierce crop	color jitter	grayscale	rotation
Top-1 accuracy	82.18%	80.00%	78.91%	80.36%	82.55%	81.09%

Contrastive Learning. SCL [19] is a popular supervised contrastive learning framework. However, as pointed out by [43], its loss function is unsuitable for long-tailed distribution. We additionally test a supervised contrastive learning framework designed for long-tail distribution [43] and find its result also limited. Therefore, we directly use the standard contrastive learning loss 1.

Multi-task Classification Module. Figure 4 demonstrates the Pipeline of our classification module. Built upon a standard encoder (*e.g.,* ResNet), our model jointly solves two tasks: (1) the label-concerned contrastive learning task discussed above and (2) the standard classification task. Given an input mini-batch $\mathcal{X}$ and two augmentation distribution $\mathcal{T}_{strong}$, $\mathcal{T}_{weak}$, we draw three samples for each $x \in \mathcal{X}$. The first two samples are drawn from $\mathcal{T}_{strong}$ and are used to calculate contrastive

loss ($\mathcal{L}_{\text{SupCL}}$), while the third sample is drawn form $\mathcal{T}_{weak}$ and is used to calculate classification loss ($\mathcal{L}_{\text{CE}}$).

The overall loss function of our classification module is formulated as follows:

$$\mathcal{L}_{\text{total}} = \lambda \cdot \mathcal{L}_{\text{SupCL}} + \mathcal{L}_{\text{CE}}, \tag{2}$$

where λ is a weight parameter.

5 Pose Estimation

5.1 Primates Pose Estimation

MacaquePose. MacaquePose [21] provides pose annotation of primates. MacaquePose focuses on 2D pose estimation of animals, including species such as chimpanzees, macaques, and others. The images are partly taken from zoo shots and partly from data on the internet. The total number of images, including annotations, is 13,083. Both in terms of number and species diversity, the objective of training in pose estimation has been achieved.

Pre-trained Model. We utilize the top-down methods. We perform object detection first, followed by single-object pose estimation given object bounding boxes. Instead of directly estimating keypoint coordinates, the pose estimator will produce heatmaps representing the likelihood of being a key point. We use the pre-trained model provided by mmpose [7] for testing and fine-tuning.

Network Architecture. We used ResNet and HRNet [35] as the network architecture of the algorithm, and tried different depths for ResNet and different widths for HRNet. Our pipeline is shown in Figure 5.

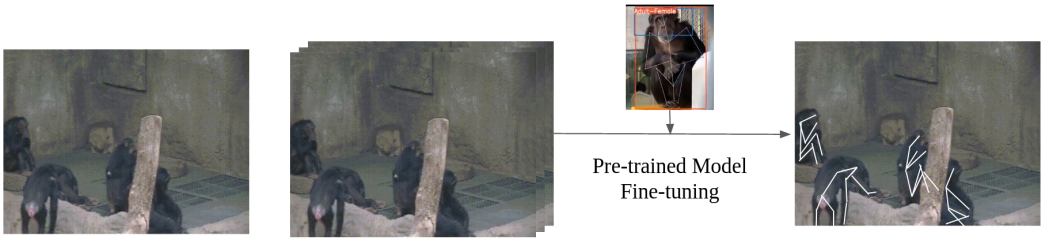


Figure 5: Pipeline of primates pose estimation.

5.2 Human-to-Primates Pose Estimation

As the application scenario for human pose estimation is much broader than for primate pose estimation, the dataset for human pose annotation is much larger, and related work is increasing. We may therefore be able to learn primate pose estimation with the help of fine-tuning, Few-shot, or transfer learning based on a human 2D pose estimation model. The pipeline is shown in Figure 6.

Due to time constraints, we have only tried this method initially. We first obtain a pre-trained 2d human pose estimation model and fine-tune it on our dataset. The AP achieved when putting the pre-trained model directly into our test set is only 27.5%, while after fine-tuning, the AP reaches 31.3%.

We found that the accuracy of using the pre-trained model directly is very low and has been improved by fine-tuning. However, there is a huge difference between human and animal appearance, and the overall success rate is still unsatisfactory.

6 Experiments

6.1 Detection

Baseline. We use yolov5s as the original architecture. Table 2 is the test results of the model after 300 epochs of training.

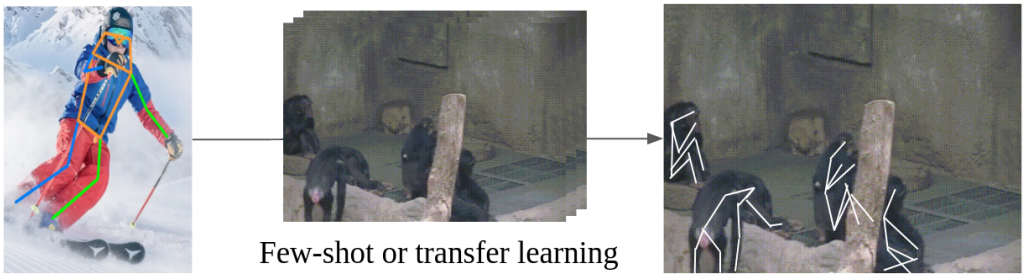


Figure 6: Pipeline of Human-to-Primates pose estimation.

Table 2: Test results of the original model

Architecture	Training Epochs	P	R	mAP_{50}
yolov5s	300	0.874	0.786	0.871

Different numbers of training epochs. We change the number of training epochs according to the optimization method in section 3.3. Table 3 shows the results before and after optimization. The optimization of the number of training epochs has significantly improved the performance of the model. The problem of over-fitting is partly alleviated.

Different model sizes. We optimized the size of the model according to the method in section 3.3. Table 4 is the test results of the models with the architecture of yolov5s, yolov5m and yolov5l on the test set. It can be seen that increasing the size of the model can improve the performance of the model. Specific information of different models are shown in Table 5.

6.2 Idenfication

Baselines. We use vanilla supervised learning with no data augmentation as the baseline. For fair comparisons, all models share the ResNet-18 architecture. Training lasts for 50 epochs with Adam optimizer and cosine learning rate scheduler. The initial learning rate is set to 1e-4.

Implementation Details. In our method, $\mathcal{T}_{weak}$ consists of mild resized crop and horizontal flip, while $\mathcal{T}_{strong}$ consists of the resized crop, horizontal flip, color jitter, and random rotation. The projection head is an MLP with one hidden layer.

Benchmarking Results and Ablations. Table 6 shows the benchmarking results. Compared with the no augmentation baseline, our method improves the identification accuracy by 3.27% (from 82.18% to 85.45%). Under the contrastive learning paradigm, we compare our approach (multi-task, MT) with pretrain+finetune (PF). Our multi-task design yields a performance gain of 2.18%. These results justify the superiority of our proposed method.

6.3 Pose Estimation

Metrics. Keypoint Detection uses a metric called Object Keypoint Similarity (**OKS**) to quantify the closeness of the predicted keypoint-location with the ground-truth keypoint. This metric ranges between 0 and 1. The closer the predicted key point to the ground truth, the closer will OKS approach 1. The formula is as follows:

$$OKS = exp(-\frac{d_i^2}{2s^2k_i^2})$$

Table 3: Test results of the models before and after the optimization of number of training epochs

Architecture	Training Epochs	P	R	mAP_{50}
yolov5s	300	0.874	0.786	0.871
yolov5s	90	0.886	0.821	0.887

Table 4: Results of detection of models with different sizes

Architecture	Training Epochs	P	R	mAP_{50}
yolov5s	90	0.886	0.821	0.887
yolov5m	67	0.865	0.812	0.889
yolov5l	40	0.852	0.845	0.890

Table 5: Sizes of different models

Architecture	params(M)	FLOPs(B)
yolov5s	7.2	16.5
yolov5m	21.2	49.0
yolov5l	46.5	109.1

Where d_i is the Euclidean distance between the predicted and ground truth, s is the object's scale, and k_i is a constant for a specific key point.

OKS is a metric value for objects. After the calculation of OKS, it plays the same role as the IOU inside the target detection so that we can set the threshold filter, then we get our Average precision(AP) and average recall(AR) metrics: AP, AP_{50}, AP_{75}, AR, AR_{50}. The result can be found in Table 7.

Network Architecture. We used ResNet and HRNet [36] as the network architecture of the algorithm, and tried different depths for ResNet and different widths for HRNet.

Ablation Study. We compare our method with some ablation methods, including *Ours w/o Augmentation* and *Ours w/o Fine-tuning*. As shown in Table 8, we can observe that our method is better than the other two methods.

7 Conclusion

In this report, we explore three representative tasks on primates. We reproduce multiple baselines and present novel approaches towards these tasks. Extensive experiments demontrate the superiority of our methods.

8 Acknowledgements

Zebin Yang, Rundong Luo, and Yiran Geng are responsible for the detection, identification, and pose estimation. The abstract, introduction, conclusion, and rebuttal are written by Rundong, while Yiran is in charge of slides and demos. Although the experiments are written by each author separately, they discuss and brainstorm a lot throughout the research process. Finally, we sincerely thank Prof. Yixin Zhu, Dr. Siyuan Huang, and TA for their precious help.

Table 6: Benchmarking results and ablations for the classification task.

Method \ Augmentation	weak	strong	strong&weak (**proposed**)
Baseline (vanilla supervised)	82.18%	77.09%	N/A
Multitask (**proposed**)	82.91%	77.45%	**85.45%**

Method	supervised	contrastive-PF	contrastive-MT (ours)
Top-1 accuracy	82.18%	83.27%	**85.45%**

Table 7: Average precision(AP) and average recall(AR) of pose estimation with different network architectures and network parameters.

Architecture	AP	AP_{50}	AP_{75}	AR	AR_{50}
ResNet-50	0.489	0.672	0.629	0.527	0.664
ResNet-101	0.500	0.638	0.602	0.515	0.668
ResNet-152	0.499	0.667	0.625	0.552	0.672
HRNet-w32	0.510	0.655	0.620	0.560	0.671
HRNet-w48	0.514	0.666	0.614	0.560	0.665

Table 8: Ablation Study for average precision(AP) and average recall(AR) of pose estimation.

Method	AP	AP_{50}	AP_{75}	AR	AR_{50}
Ours	0.514	0.666	0.614	0.560	0.665
Ours w/o Augmentation	0.496	0.649	0.602	0.530	0.666
Ours w/o Fine-tuning	0.392	0.550	0.512	0.448	0.560

References

[1] Abdallah Benzine, Florian Chabot, Bertrand Luvison, Quoc Cuong Pham, and Catherine Achard. Pandanet: Anchor-based single-shot multi-person 3d pose estimation. In *CVPR*, 2020. 3

[2] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. *NeurIPS*, 2019. 3

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[4] Xianjie Chen and Alan L Yuille. Parsing occluded people by flexible compositions. In *CVPR*, 2015. 3

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 3

[6] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *ECCV*, 2020. 2

[7] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. `https://github.com/open-mmlab/mmpose`, 2020. 3, 6

[8] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *ICCV*, 2017. 2

[9] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, 2017. 3

[10] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 1

[11] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014. 3

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[14] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016. 2

[15] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. In *ECCV*, 2016. 3

[16] Redmon Joseph, Divvala Santosh, Girshick Ross, and Farhadi Ali. You only look once:unified, real-time object detection. *arXiv*, 2016. 1, 2

[17] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *ICLR*, 2020. 2

[18] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2

[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 2, 5

[20] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 2019. 3

[21] 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*, 2021. 1, 3, 6

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 2

[24] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *ICCV*, 2019. 3

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 2

[26] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv*, 2017. 2

[27] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*, 2012. 3

[28] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, June 2016. 3

[29] Girshic Ross, Donahue Jeff, Darrell Trevor, and Malik Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 1

[30] Daniel Schofield, Arsha Nagrani, Andrew Zisserman, Misato Hayashi, Tetsuro Matsuzawa, Dora Biro, and Susana Carvalho. Chimpanzee face recognition from videos in the wild using deep learning. *Science Advances*, 2019. 1

[31] Ren Shaoqing, He Kaiming, Girshick Ross, and Sun Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv*, 2016. 2

[32] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3

[33] Kreiss Sven, Bertoni Lorenzo, and Alahi Alexandre. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE TITS*, 2021. 3

[34] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *arXiv*, 2019. 3

[35] Juergen Gall Umar Iqbal. Multi-person pose estimation with local joint-to-person associations. In *ECCVW*, 2016. 3

[36] 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. *arXiv*, 2019. 8

[37] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *ECCV*, 2020. 3

[38] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 3

[39] Zhou Xingyi, Wang Dequan, and Krähenbühl Philipp. Objects as points. In *arXiv*, 2019. 3

[40] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, 2017. 2

[41] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv*, 2021. 2

[42] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, 2021. 2

[43] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *CVPR*, 2022. 2, 5