The hitchhiker's guide to endangered species pose estimation

Jakub Straka[1], Marek Hruz[1], Lukas Picek[1,2]
[1]University of West Bohemia, [2]INRIA
strakajk,mhruz@kky.zcu.cz, picekl@kky.zcu.cz/lpicek@inria.cz

Abstract

Preserving endangered species is a critical component of maintaining a balanced and healthy ecosystem. Animal pose, especially for rare animals, allows an understanding of various aspects of biology and ecology, including but not limited to individual animal behavior analysis and study of migration patterns. Using the small-scale dataset from (i.e., red-list species) monitoring efforts of Eurasian lynx (Lynx lynx), we provide a comprehensive guide to a simple yet effective 2D pose estimation suitable for endangered species. We showcase the contribution of a variety of methods and their influence on the performance in terms of AP, $AP_{0.75}$, $AP_{0.85}$, and $PCK_{0.05}$. Our experiments provide a hitchhiker's guide to (i) pre-trained model selection, (ii) model pre-training and fine-tuning, (ii) augmentation strategies, (iii) training hyper-parameters settings, (iv) number of required real data, and (v) use of synthetic data. Using all the bells and whistles and HRNet-w32, we achieved 0.855AP and 0.936$PCK_{0.05}$ lowering the relative error of a pre-trained model by more than 50%. Last but not least, we have developed a system for photorealistic synthetic camera trap data generation. The system is available at: https://github.com/strakaj/Synthetic-animal-pose-generation.git.

1. Introduction

Ensuring the survival of endangered species is vital for sustaining a harmonious and thriving ecosystem. Monitoring the population and migration of endangered species has been at the center of the attention of biologists for a long time. Non-intrusive tracking devices such as camera traps are preferred to other devices such as collars or chips. However, processing the data from cameras by hand is time-consuming and tedious. Modern AI-powered systems can be helpful in many ways, such as filtering empty camera images [6], classifying the observed species [25], identifying individuals [14], or estimating their pose [31].

Animal pose plays a crucial role in studying and understanding various aspects of biology and ecology, e.g., animal behavior [10, 13, 22], migration patterns [11, 17], that allow for better conservation and endangered species protection [27, 35]. In general, pose estimation is a challenging task for several reasons, such as the inherent flexibility of limb joints and the potential for body parts to be obscured. In human pose estimation, a model trained on one dataset can be easily transferred to different data as humans in different scenarios are still similar in size and proportions. There is a large number of animal species and not all species are represented in existing datasets. Therefore, in some cases, it is necessary to transfer a model trained for detection on animals of one species to animals of different species which can differ in size, proportionality, and flexibility of limbs making this task more challenging. In recent years, animal pose estimation has received more attention with newly proposed datasets with a large range of species [22, 32, 33]. Still, rare and endangered species are

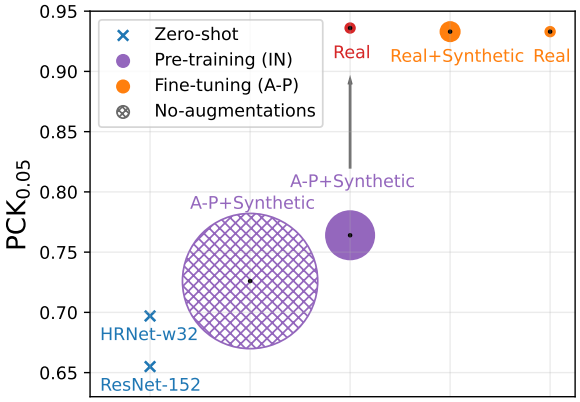


Figure 1. **Results summary for 2D pose estimation of endangered species *Lynx lynx* with limited data**. We showcase that pre-training on larger datasets leads to overfitting, and any amount of real data leads to better performance. Overall, the best approach is to pre-train on domain-related data enriched with synthetic data and fine-tune it with any available real data. The size of the circles is proportional to the number of training data; the largest circle has 14000 images, and the smallest includes 1000 images. A-P stands for Animal-Pose dataset and IN for ImageNet-1k dataset.

omitted or represented sparsely in these datasets. This is due to the nature of their scarcity, which makes them difficult to capture on a camera. Existing pre-trained models for animal pose estimation suffer from the poor generalization ability to unseen and endangered (i.e., rare) animal species. In such cases, one of the solutions can be the use of synthetic data.

Arguably, the photorealisticity of generated synthetic data can reduce the domain gap between real and synthetic data. Studies [26, 29] on other computer vision tasks (semantic segmentation) showed that highly realistic synthetic data can outperform less realistic synthetic datasets. However, such datasets may face challenges in terms of lower data diversity due to the difficulty of creating detailed scenes. Real data encompass images from diverse environments with varying weather and lighting conditions, leading to significant appearance variations at the same location over time. Synthetic data generation pipeline should address this issue. Another challenge is the use of realistic models of animals in simulation. Real animals of one species can vary in size, texture, and proportions. Simulation should also capture animals in various poses.

Synthetic data offers certain advantages over real data. Annotation of real data may be challenging primarily due to occlusion and photo quality. This can lead to inaccurate position of keypoints or only partial annotation of the pose. Simulation of synthetic data provides the advantage of precise control over keypoint positions, ensuring their accuracy. Furthermore, generating data for different animal species becomes straightforward by merely switching between animal models in the simulation process.

The main contributions of this work are: 1. We provide an all-encompassing guide on how to handle scarce data of endangered species for pose estimation. 2. We explore the effect of synthetic data on the quality of pose estimation by creating a high-detailed synthetic image dataset of the target animal - *Lynx lynx*. 3. We analyze the metrics that measure pose estimation accuracy and propose a stricter keypoint variance value so that the accuracy measure is more intuitive. 4. We measure the influence of pre-training and fine-tuning of pose estimation approaches on different data.

2. Related Work

This work is related to several topics as it provides a guideline for endangered species 2D pose estimation in scenarios with limited data. Therefore, in this section, we describe the state-of-the-art in these different topics, i.e., animal pose estimation, utilization of synthetic data, and the available *real* data.

2.1. 2D Pose Estimation

2D human or animal pose estimation is usually done using the skeletal pose estimation, where the pose is defined as a graph with 2D positions of joints as nodes that are connected by bones represented as vertexes of the graph. There are two major approaches – *Top-Down* and *Bottom-Up* – to detect the 2D positions of joints (alternatively bones). In both, the pose estimation model comprises two parts: (i) a backbone (usually a pre-trained CNN- or Transformer-based model) and (ii) a segmentation head that directly regresses the location or uses multiple transposed convolutions that result in a heatmap for each key point. The position of the key point is then obtained as the pixel's position with the maximum value in the corresponding heatmap.

The **Top-Down approach** is characterized by first detecting instances of the objects and then estimating the pose for each instance individually. Examples of Top-Down methods include the stacked hourglass model [21] or the HRNet model [24], which preserves the high resolution of the input throughout the network. Simple but strong baselines for Top-Down pose estimation are presented in the Simple Baseline paper [30]. Recently, following the success of the Transformers in NLP [28] and CV [8], the Transformers has also been adopted to pose estimation [31].

Differently, the **Bottom-Up approach** detects all the joints of all the object instances directly on the entire input image. Such approaches include OpenPose [3], which joins the detected joints into skeletal graphs using the graph-matching method, and HigherHRNet [4], which learns scale-aware representations using high-resolution feature pyramids.

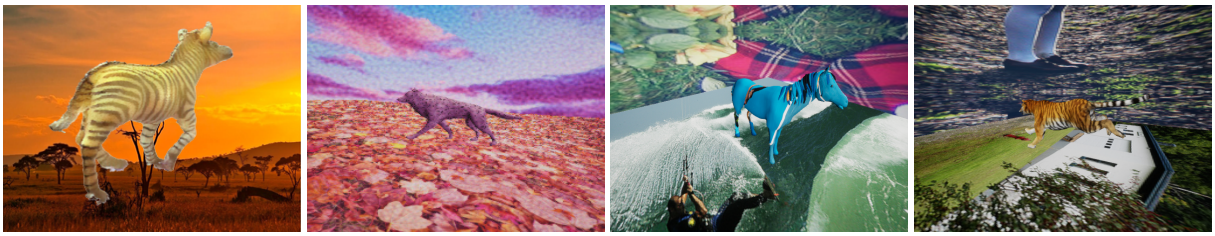
Animal Pose Estimation is adopted from the Human Pose Estimation in the sense of methods and techniques, but a different skeletal model is used. The best-performing methods on the animal pose benchmark datasets (e.g., Animal-Pose [2] and AP-10k [33]) are the Transformer-based ViTPose model [31] and the CNN-based models HRNet-w32 and HRNet-w48[1] [24].

2.2. Learning with Synthetic Data

Synthetic data are a promising way of addressing the lack of labeled real data, particularly in the problem of animal pose estimation. There are several approaches to leveraging synthetic data. In [15, 20], the authors trained models on synthetic data, which are then used to generate pseudo-labels for real data. This approach assumes a significant number of real images of target species, which may not be viable for rare and endangered species. The study by Mu et al. [20] also showed that the model trained exclusively on their synthetic data did not perform well on real data.

A study by Shooter et al. [23] proposes a synthetic dataset of dogs. Images are generated by creating a simple 3D scene with a dog. A pose of the animal model is

[1] 32 and 48 represent the width of the high-resolution sub-network in the last three stages.



Jiang et al. [12] Shooter et al. [23] Mu et al. [20] Mu et al. [20]

Figure 2. **Examples of generated synthetic data**. Recently published methods do not provide a photo-realistic output. Methods that place a 3D model into a static 2D background [12] do not account for interaction and occlusions caused by background. More advanced methods use a 3D squared space [15, 20, 23], this provides more variability in environment appearance and more realistic model placement but still does not account for interaction and occlusions caused by background.

generated using the Adaptive Neural Network proposed by Zhang et al. [34]. Multiple components of the scene are randomly selected e.g. background image and camera position. The models trained exclusively on synthetic data performed poorly. Only after combining synthetic data with a portion of training data from StanfordExtra [1] models surpassed the performance of models trained only on real data.

Similarly, Jiang et al. [12], focused on generating synthetic data and training on a combination of synthetic and real images. Synthetic data are generated by composing real background images and rendered 3D models. The pose of the model is generated through a variational autoencoder. Additionally, the style adaptation model StyTr^2 [7] is used to match the style of the background and rendered animal. Results showed that synthetic data in combination with real data can slightly improve performance over the models trained only on synthetic data.

The described methods show that there is potential for the use of synthetic data when real data are scarce. However, the photo-realism of the synthetic data generated by prior works remains questionable. See Figure 2 for examples. In this work, we push the quality of the synthetic data to the limit to see whether it has any considerable effect on the quality of predictions on real data.

2.3. Animal Pose Datasets

Data are crucial for supervised learning of animal pose. There were several single-species datasets proposed [1, 16, 19] in the past. Ideally, for endangered species, the model would be pre-trained on a dataset with a large variety of animal species and then fine-tuned on the smaller species-specific dataset. One of the first attempts to create a multi-species dataset is the Animal-Pose [2] dataset which contains 5.5k annotations for 5 common animal species. Each animal is annotated with a maximum of 20 keypoints. In [33] the authors proposed the AP-10k dataset which addresses the low diversity of species in [2]. The dataset contains 13k instances of 54 species from 23 families annotated with a maximum of 17 keypoints. The distribution of species in the dataset is not uniform, rare species are represented less, which corresponds to their occurrence in nature. Another diverse dataset is APT-36k [32]. The dataset is created from video clips and focuses on pose estimation and tracking. The dataset contains 53k instances of animals of 30 species. The same 17 keypoints as in [33] are used.

3. Datasets

We focus on the 2D pose estimation of an endangered species *Lynx lynx*. In this section, we describe (i) a small-scale camera-trap dataset collected within central Europe and (ii) a newly developed pipeline for realistic synthetic data generation of any endangered species.

3.1. Central Europe Lynxes

The Central Europe Lynxes dataset originates from the camera trapping project that focuses on the monitoring and conservation of *Lynx lynx* in Central Europe and has been running for almost 15 years. More details about the acquisition process and details of manual identification of lynx individuals based on a comparison of coat patterns, particularly on the hind limbs, forelimbs, and flanks, are available in a study by Dula et al. [9].

From the provided unstructured data, we have built two datasets – for testing and training – that originate from two geographically distinct regions – Šumava National Park (1969 images) and Javorníky (1148 images) – with different backgrounds and camera traps to allow better cross-geographical performance evaluation. In all our experiments and ablations, the Šumava dataset is used for testing, and the dataset from Javorníky is used for training. While constructing the datasets, we excluded the images with more than one individual and hardly visible or heavily occluded Lynxes. All Lynx lynx instances were annotated with 20 keypoint skeletons based on the physiology of the *Felidae* species and to allow compatibility with the Animal-Pose dataset. Besides the keypoints, a bounding box is provided.

3.2. Synthetic Data Generation

Synthetic data have the potential to increase performance in scenarios with limited data, such as 2D pose estimation of endangered species. Given the fact that outputs of existing approaches for synthetic animal pose estimation are far from realistic [12, 23] (see Figure 2), we have proposed and developed a new pipeline that uses a game engine (i.e., Unity) capable of producing highly realistic synthetic samples of any species. The comprehensive overview of the proposed pipeline, which consists of a description of the Environment, Animal, and skeletal models, is provided below.

Environment: To generate highly realistic environments (i.e., scenes), we have used freely available assets (e.g., terrain, trees, logs, etc.) from the Unity's Book Of The Dead. Inspired by the real environments that are available in the data, we have created four photo-realistic scenes with dynamic changes in real environments (e.g., tree type, grass, snow). Different environment variants can be found in the supplementary material. We show the comparison between real and synthetic scenes in Figure 3.

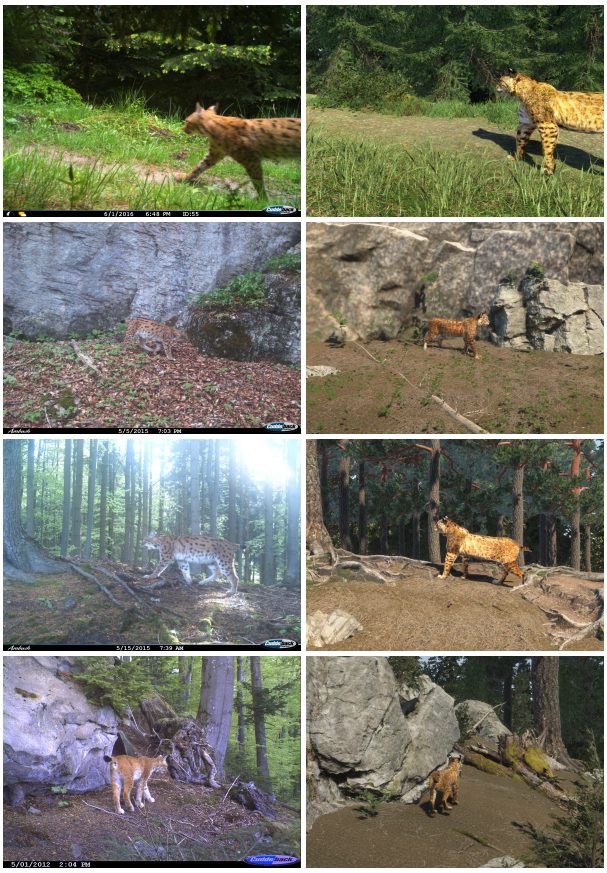


Figure 3. **Selected synthetic data samples**. Inspired by the real camera trap views, we have created four highly realistic scenes. All scenes are made publicly available for further use.

Animal model: The main asset of our approach is the detailed synthetic *Lynx lynx* textured model. On the negative side, the model has just one walking animation. Therefore, we have developed a sitting animation in addition to the existing walking animation. We use the same skeleton model as in the Animal Pose dataset with the 20 keypoints to allow direct comparison with real data. In addition to keypoints, we also save a bounding box, which is obtained based on the mesh of the model.

Data generation pipeline: In the data generation process, we aimed to mimic the actual process of capturing photos through camera traps. First, we manually select four points that define the trajectory within each scene where the animal model will walk. Along the walk, the *Lynx lynx* model "stops" n times. In each "stop" several adjustments to the environment (e.g., tree type, grass, snow), camera viewpoint, and animation and rotation of the animal modes are made. Subsequently, two data samples are saved. First with the pre-stop conditions and second after the adjustments.

4. Evaluation Metrics

Commonly used metrics for 2D pose estimation are *Percentage of Correct Keypoints* (PCK) and Average Precision (AP) based on *Object Keypoint Similarity* (OKS). In this section, we briefly describe both methods and further evaluate their weaknesses, strengths, and suitability.

Percentage of correct keypoints measures if the distance between prediction and ground truth is less than the given threshold. Typically, the threshold is determined as a fraction of the animal's size. This implies that every key point must be predicted with equal accuracy regardless of their ambiguity. For instance, the elbow is more ambiguous than the eye, but the same level of accuracy is expected.

Object Keypoint Similarity is used in the calculation of AP to measure the similarity between prediction and ground truth value. The keypoint similarity (KS) is computed as $KS = \exp\left(\frac{-d^2}{2s^2 2\sigma^2}\right)$, where d is the distance of the predicted keypoint and the ground truth, s is the scale of the detection, and σ is the standard deviation of redundantly annotated per-keypoint annotations. The OKS is then computed as the mean of visible keypoints KS. A keypoint is considered correctly predicted if its KS value is larger than a selected KS threshold.

Average precision calculates the percentage of correctly predicted keypoints. Typically, AP is evaluated on 10 evenly distributed OKS levels starting from 0.5 and ending at 0.95. For the low values of OKS, predictions can lie further from the ground truth to be considered correct. Compared to PCK, AP is able to control the size of the threshold with per-keypoint standard deviation values. This allows for less strict thresholds for ambiguous keypoints.

Figure 4. **AP thresholds in Animal-Pose dataset**. An illustration of AP thresholds for selected keypoints using three different OKS values (e.g., 0.5, 0.85, and 0.95).

Metric evaluation: To understand the quality of the pose estimation intuitively, one has to understand the different parameters of individual metrics. At first, we used σ values from the Animal-Pose dataset introduced by MMPose [5] that were inferred from the original COCO [18] values for human pose estimation. After a brief evaluation of various thresholds, we observed that at low OKS levels, predictions were considered correct even if they were predicted on incorrect parts of the animal body (see Figure 4).

We argue that AP evaluated on low OKS levels (e.g., 0.5–0.75) with standard deviation values proposed for Animal-Pose is not meaningful for model comparison and can be misleading. Based on our observations, we adjusted the σ values[2] to more precisely measure the precision of the model. Figure 5 depicts a comparison of Animal-Pose OKS thresholds and OKS thresholds with our adjusted σ values. We adjusted σ values with two goals in mind. The least strict thresholds of two keypoints should overlap as little as possible and thresholds should extend beyond the body of the animal as little as possible. In Figure 6 we compare APs and $\text{PCK}_{0.05}$ with our modified values of σ.

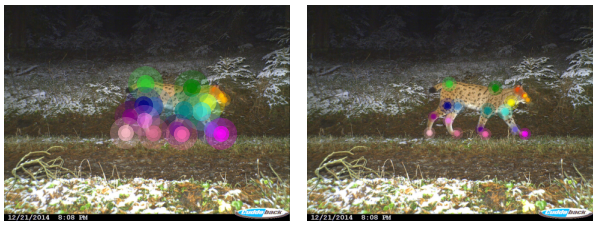


Figure 5. **Per-keypoint standard deviation comparison.** In the image on the left are illustrated OKS thresholds with standard deviation proposed for the Animal-Pose dataset. In the image on the right are our adjusted values of the standard deviation.

[2]See supplementary material for details.

Figure 6. **Proposed AP threshold**. With our adjusted σ values even the least strict threshold does not excessively overlap with other parts of the animal. AP thresholds are at 0.5 and 0.85 OKS.

5. Methods

In order to establish guidelines for *simple yet effective* 2D pose estimation of endangered species in a scenario with limited data, we performed multiple experiments. The experiments are tailored to provide the reader with a guide to (i) pre-trained model selection, (ii) model pre-training and fine-tuning, (ii) augmentation strategies, (iii) training hyper-parameters settings, (iv) number of required real data, and (v) use of synthetic data. In this section, we describe all relevant experiments, training strategies, and evaluations. All experiments were initialized with the same random seed and training hyperparameters to ensure fair comparison. If not stated differently, the models were optimized for 210 epochs with a mini-batch size of 64, using Adam optimizer with multistep LR scheduler.[3]

Method and backbone: Based on the state-of-the-art performance in both human and animal 2D pose estimation, we focus on the Top-Down methods in all experiments. To find the best suitable method and backbone, we evaluate the performance of different pre-trained ResNets (ResNet-50, ResNet-101, and ResNet-151) and HRNets (HRNet-w32, HRNet-w48) available in MMPose library.

Fine-tuning with synthetic data: As proposed in related work, synthetic data might have a positive influence on the performance of animal pose estimation methods. To test this scenario with limited data, we use the publicly available data from the Animal-Pose dataset and our generated synthetic data. We started training from the model pre-trained on ImageNet. We always used all Animal-Pose

[3]The initial learning rate was $5e^{-4}$ which was reduced by 90% after the 80% epoch and again after the 95% epoch.

data and gradually added synthetic data from 0 to 10k images to evaluate the influence on performance. Smaller versions of the dataset are always selected as a subset of the larger versions and images are evenly selected from four individual scenes.

Augmentations: Up to this point, we have not addressed any post-processing of generated synthetic data. Instead of applying transformations during the generation of the data we left this procedure until the training stage, where it can be addressed through augmentations. This gave us an opportunity to assess the influence of different augmentations on performance. We selected five augmentations based on different aspects of the real data that were not sufficiently addressed in the simulation. To replicate imperfections in the camera sensor, we applied blur and Gaussian noise augmentations. Different lighting conditions are simulated as HSV shift and random brightness and contrast augmentation. Because some of the real data are taken at night we also added augmentation that converts images to grayscale.

Hyperparameters fine-tuning: Optimally selected hyperparameters are crucial for training. To ensure that the parameters are optimal we performed a hyperparameter sweep over learning rate and number of epochs. The learning rate was selected from the set: $\{5e^{-4}, 1e^{-4}, 1e^{-5}\}$ and the number of epochs from the set: $\{50, 100, 210\}$. Models were trained on a combination of 1k synthetic data and the Animal-Pose dataset.

Fine-tuning with real data In this experiment, we measured the performance of models trained on real lynx data. We considered the scenario where real lynx data are available but limited. We tested two cases, in the first case, the model was trained from the ImageNet checkpoint with a combination of Animal-Pose data and real lynx data. In the second case, we trained the model from the Animal-Pose checkpoint only on real data.

Fine-tuning with synthetic and real data: Based on the related work relying only on synthetic data is not enough to outperform models trained on real data. However, combining real and synthetic data has the potential to improve results. We explored two strategies for combining real and synthetic data. In the first strategy, we combined real lynx data with synthetic data in different proportions and trained models from the Animal-Pose checkpoint.

In the second strategy, we first trained the model from the ImageNet checkpoint on a combination of the Animal-Pose dataset and 1k synthetic data. Then we continued training the model with real data. In this way, the model can learn prior knowledge about the lynx pose from synthetic data and then fully leverage information from real images.

6. Results

We performed a series of experiments in which we showed different possible utilizations of synthetic and real data. In this section, we will provide the results of the experiments. As mentioned earlier, we have two datasets at our disposal that contain real images of the *Lynx lynx* species. For all our experiments, we used the dataset from Šumava National Park (1969 images), for evaluation. In experiments where real lynx data were used for training, we used the dataset from Javorníky (1148 images).

Backbone selection: Evaluating available backbones, e.g., ResNet-50, ResNet-101, ResNet151, HRNet-w32, and HRNet-w48 in the zero-shot scenario, shows that HRNet models achieve a superior performance to ResNets. The best model, HRNet-w32, reduced the ResNet-152's – a model with 50% more parameters – relative error by 21.0%. Interestingly, HRNet-w32 outperformed the larger HRNet-w48 by a considerable margin. This may be due to the relatively low number of training data. This also follows from the results on the other datasets provided by MMPose. HRNet-w32 performed better when trained on the Animal-Pose dataset, however, HRNet-w48 achieved superior performance when trained on the larger AP-10k dataset. Nevertheless, HRNet-w48 has almost twice the number of FLOPS. We provide the comprehensive performance overview in Table 1. Based on our evaluation, the most viable method for animal pose estimation is HRNet-w32.

Backbone	AP	$AP_{0.75}$	$AP_{0.85}$	$PCK_{0.05}$	Params	FLOPS
ResNet-50	0.251	0.087	0.010	0.613	*25.6*	*4.1*
ResNet-101	0.309	0.113	0.005	0.641	*44.6*	*7.9*
ResNet-152	0.333	0.157	0.010	0.655	*60.2*	*11.6*
HRNet-w32	0.403	0.277	0.013	0.697	*41.2*	*9.0*
HRNet-w48	0.379	0.199	0.010	0.678	*77.5*	*17.4*

Table 1. **Zero-shot performance of selected Animal-Pose pretrained backbones**. The performance is evaluated on real *Lynx lynx* data using standard AP and PCK metrics. Parameters of models are given in millions and floating operations in gigaFLOPS.

Fine-tuning with synthetic data: Using available Animal-Pose data enriched with an increasing number (0, 10, 100, 1k, 10k) of synthetic data showed a constant improvement trend in performance. The performance peak (0.475AP), where the performance started decreasing, i.e., the model was overfitting, was achieved at 10k images. We show the effect of synthetic data on performance in Figure 7. Interestingly, using only the AP dataset led to slightly better results than provided in the MMPose documentation, most likely due to different random seeds.

All proposed augmentations were beneficial for performance. The most beneficial were augmentations that al-

tered the lighting conditions of the image. Random brightness and contrast adjustment were the most beneficial from the individual augmentations and improved AP compared to baseline by 7.8%. By using all augmentations together AP improved by an additional 3.2%. In Table 2 we report ablation on augmentations. Based on the ablation use of all proposed augmentations is beneficial.

Based on the augmentation results we trained the model again with an increasing number of synthetic images, but this time with all proposed augmentations. The model trained with 1k synthetic images outperformed the model trained with 10k images. AP of the model improved from 0.464 to 0.516. This could be attributed to the fact that when trained without augmentations model was more prone to overfiting. The model trained with 10k images did not improve as much. Performance increased from 0.475 to 0.494AP. The results of both experiments are compared in Figure 7. We concluded that augmentations are crucial for training and can dramatically decrease the number of necessary synthetic images.

noise	*blur*	*to gray*	*hsv*	*brightness*	**AP**	**AP**$_{0.75}$	**AP**$_{0.85}$	**PCK**$_{0.05}$
✓					0.466	0.449	0.030	0.732
	✓				0.476	0.470	0.055	0.738
		✓			0.490	0.511	0.063	0.746
			✓		0.494	0.518	0.059	0.745
				✓	0.500	0.526	0.064	0.751
✓	✓	✓	✓	✓	0.516	0.554	0.102	0.764
Baseline – no augment.					0.464	0.453	0.019	0.728

Table 2. **Ablation on augmentations.** We evaluate various augmentations and their influence on the real test data performance using synthetic data and HRNet-w32.

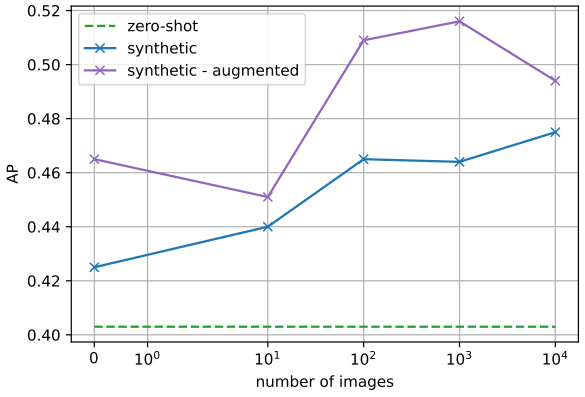


Figure 7. **Ablation on synthetic data pre-training**. We enrich the Animal-Pose dataset with a different number of *Lynx lynx* synthetic data and pre-train the HRNet-w32 model for 210 epochs.

Hyperparameters: Results of the sweep showed, that generally, models trained with higher learning rates and a lower number of epochs achieved better results. The best results were achieved with a $1e^{-4}$ learning rate and 100 epochs. The model trained with the best setting improved its performance from 0.464 to 0.486 AP. When trained with augmentations model improved from 0.516 to 0.523 AP.

Fine-tuning with real data We evaluated two scenarios where the HRNet-w32 model was fine-tuned from the ImageNet checkpoint and the Animal-Pose checkpoints using an increasing number (0, 10, 100, 1k) of real data (i.e., Javorníky), for all the cases. All experiments were performed with augmentations and improved training strategy. The model fine-tuned from the Animal-Pose checkpoint achieved a better performance in terms of AP surpassing the model fine-tuned from ImageNet. Notable are the results of the models trained with a low number of images. The model trained even with 10 real images outperformed the best model trained with synthetic images by 21.0% AP. The model trained with 1k real images achieved the highest performance, with an AP of 0.849. A comprehensive result of the two scenarios can be found in Table 3. Based on the results it is clear that the most beneficial for performance is to start training from checkpoint trained on a general animal dataset with as many real images as possible.

A-P	**Real**	**Checkpoint**	**AP**	**AP**$_{0.75}$	**AP**$_{0.85}$	**PCK**$_{0.05}$
✓	0	ImageNet	0.431	0.337	0.012	0.719
✓	10	ImageNet	0.481	0.463	0.035	0.746
✓	100	ImageNet	0.710	0.821	0.577	0.870
✓	1000	ImageNet	0.837	0.906	0.799	0.927
✗	10	Animal-Pose	0.626	0.725	0.407	0.826
✗	50	Animal-Pose	0.739	0.833	0.658	0.881
✗	100	Animal-Pose	0.769	0.845	0.704	0.895
✗	500	Animal-Pose	0.829	0.905	0.790	0.925
✗	1000	Animal-Pose	0.849	0.916	0.810	0.933

Table 3. **Effect of a number of real images on performance.** We trained HRNet-w32 on different combinations of real data. A-P indicates if the Animal-Pose dataset was used for training. In all cases was more beneficial to train model from Animal-Pose checkpoint with real data.

Fine-tuning with synthetic and real data: We measured the performance of the model fine-tuned from the Animal-Pose checkpoint with a combination of 1k synthetic data and a different number of real lynx data. The best performance achieved model fine-tuned with 1k real images and 1k synthetic images. However, the results were very close to the model trained without synthetic data, the only improvement was in AP$_{0.85}$ which improved from 0.810 to 0.821. With less real images performance decreased compared to

models trained with no synthetic data. In Table 4 is a summary of the results. This leads us to the conclusion that it is more beneficial to fine-tune models only with real images from the target domain. This is also supported by results in Table 3.

Synthetic	Real	AP	$AP_{0.75}$	$AP_{0.85}$	$PCK_{0.05}$
1000	0	0.335	0.164	0.000	0.651
1000	10	0.515	0.560	0.110	0.767
1000	100	0.741	0.844	0.676	0.892
1000	1000	0.849	0.916	0.821	0.933
50	50	0.726	0.823	0.634	0.881
100	100	0.744	0.844	0.667	0.888
500	500	0.827	0.905	0.789	0.924

Table 4. **Effect of combination of real and synthetic data on performance.** Models are trained from the Animal-Pose checkpoint. When compared to Table 3 it is clear that synthetic data does not improve performance in the fine-tuning stage and the most beneficial is to fine-tune the model with real data.

We observed that the model achieves better results when fine-tuned from a checkpoint that was pre-trained on data from a similar domain to the fine-tuning data e.g., fine-tuning from Animal-Pose compared to ImageNet checkpoint. We evaluated models pre-trained with a combination of Animal-Pose and 1k synthetic data from the ImageNet checkpoint and then fine-tuned the models with real data. With this approach, AP improved compared to the model trained only with real data from the Animal-Pose checkpoint. Improvement was more apparent at a lower number of images except when trained with only 10 real images. The model trained with 1k real images improved from 0.849 to 0.855. In Table 5 are results for different sizes of the real lynx dataset. Based on the results, when real data are available it is most beneficial to first pre-train the model on a general animal dataset (e.g., Animal-Pose) combined with synthetic data and then fine-tune the model with real data.

Real Samples	AP	$AP_{0.75}$	$AP_{0.85}$	$PCK_{0.05}$
10	0.619	0.686	0.409	0.820
50	0.758	0.854	0.676	0.895
100	0.783	0.875	0.734	0.905
500	0.838	0.907	0.809	0.931
1000	0.855	0.917	0.820	0.936

Table 5. **From synthetic data pre-training to real data fine-tuning**. We fine-tune the HRNet-w32 model pre-trained on the Animal-Pose dataset enriched with 1k of synthetic images on real data, i.e., Javorníky.

7. Conclusion

In this study, we focused on the 2D pose estimation of the Eurasian lynx, a red-listed endangered species, with limited data. We provide a complete guide to 2D pose estimation suitable for any *Felidae* species. With our comprehensive analysis, we demonstrate the impact of diverse methods on performance using standard metrics, e.g., AP, $AP_{0.75}$, $AP_{0.85}$, and $PCK_{0.05}$. Throughout our experiments, we provide insights into several key factors that influence the performance of the pose estimation model, including: (i) Selection of pre-trained models and architectures. (ii) Pre-training, fine-tuning, and augmentation strategies. (iii) Training hyper-parameter settings. (iv) Ablation on synthetic data pre-training. (v) Effect of a number of real images on performance. (vi) The effective approach for real and synthetic data combination. In a nutshell, we showcase that pre-training on larger datasets leads to overfitting, and any amount of real data leads to better performance. Overall, the best approach is to pre-train the model on domain-related data enriched with synthetic data and fine-tune it with any available real data. By employing advanced techniques and the HRNet-w32 model, we achieved remarkable results with an AP of 0.855, $AP_{0.75}$ of 0.917, $AP_{0.85}$ of 0.820, and $PCK_{0.05}$ of 0.936. This represents a significant improvement, reducing the relative error of the Animal-Pose pre-trained model by more than 50%. Furthermore, we have analyzed the established metrics for 2D animal pose evaluation and found out that the default metric parameters do not provide an intuitive insight into the quality of the estimation. Even with the most strict thresholds, the metrics can paint an inaccurate picture, giving a false impression that the pose is estimated accurately even though the individual keypoints can lie outside of the animal body or are exchanged. Thus, we adjust the keypoint variance parameters to provide a more descriptive measure. Besides, we have developed and made publicly available a system for generating photorealistic synthetic camera trap data, further enhancing the research and conservation efforts for endangered species.

8. Acknowledgments

This research was supported by the Technology Agency of the Czech Republic, project No. SS05010008 and by the grant of the University of West Bohemia, project No. SGS-2022-017. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who left the dogs out?:

3D animal reconstruction with expectation maximization in the loop. In *ECCV*, 2020. 3

[2] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9498–9507, 2019. 2, 3

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2

[4] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395, 2020. 2

[5] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 5

[6] Fagner Cunha, Eulanda M dos Santos, Raimundo Barreto, and Juan G Colonna. Filtering empty camera trap images in embedded systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2438–2446, 2021. 1

[7] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[9] Martin Duľa, Michal Bojda, Delphine B H Chabanne, Peter Drengubiak, Ľuboslav Hrdý, Jarmila Krojerová-Prokešová, Jakub Kubala, Jiří Labuda, Leona Marčáková, Teresa Oliveira, Peter Smolko, Martin Váňa, and Miroslav Kutal. Multi-seasonal systematic camera-trapping reveals fluctuating densities and high turnover rates of Carpathian lynx on the western edge of its native range. *Scientific Reports*, 11(1):9236, 2021. 3

[10] Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. Automated pose estimation in primates. *American journal of primatology*, 84(10):e23348, 2022. 1

[11] Le Jiang, Caleb Lee, Divyang Teotia, and Sarah Ostadabbas. Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities. *Computer Vision and Image Understanding*, page 103483, 2022. 1

[12] Le Jiang, Shuangjun Liu, Xiangyu Bai, and Sarah Ostadabbas. Prior-aware synthetic data to the rescue: Animal pose estimation with very limited real data. *arXiv preprint arXiv:2208.13944*, 2022. 3, 4

[13] Daniel Joska, Liam Clark, Naoya Muramatsu, Ricardo Jericevich, Fred Nicolls, Alexander Mathis, Mackenzie W Mathis, and Amir Patel. Acinoset: a 3d pose estimation dataset and baseline models for cheetahs in the wild. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13901–13908. IEEE, 2021. 1

[14] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Steffen Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, et al. Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods*, 19(4):496–504, 2022. 1

[15] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1482–1491, 2021. 2, 3

[16] Shuyuan Li, Jianguo Li, Hanlin Tang, Rui Qian, and Weiyao Lin. Atrw: a benchmark for amur tiger re-identification in the wild. *arXiv preprint arXiv:1906.05586*, 2019. 3

[17] Xiangtao Li, Jie Zhang, and Minghao Yin. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural computing and applications*, 24:1867–1877, 2014. 1

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[19] Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yuksekgonul, Byron Rogers, Matthias Bethge, and Mackenzie W Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868, 2021. 3

[20] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395, 2020. 2, 3

[21] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016. 2

[22] Xun Long Ng, Kian Eng Ong, Qichen Zheng, Yun Ni, Si Yong Yeo, and Jun Liu. Animal kingdom: A large and diverse dataset for animal behavior understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19023–19034, June 2022. 1

[23] Moira Shooter, Charles Malleson, and Adrian Hilton. Sydog: A synthetic dog dataset for improved 2d pose estimation, 2021. 2, 3, 4

[24] 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. 2

[25] Mengyu Tan, Wentao Chao, Jo-Ku Cheng, Mo Zhou, Yiwen Ma, Xinyi Jiang, Jianping Ge, Lian Yu, and Limin Feng. An-

imal detection and classification from camera trap images using different mainstream object detection architectures. *Animals*, 12(15):1976, 2022. 1

[26] Apostolia Tsirikoglou, Joel Kronander, Magnus Wrenninge, and Jonas Unger. Procedural modeling and physically based rendering for synthetic data generation in automotive applications, 2017. 2

[27] Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W Mathis, Frank van Langevelde, Tilo Burghardt, et al. Perspectives in machine learning for wildlife conservation. *Nature communications*, 13(1):792, 2022. 1

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[29] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing, 2018. 2

[30] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 2

[31] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose+: Vision transformer foundation model for generic body pose estimation. *arXiv preprint arXiv:2212.04246*, 2022. 1, 2

[32] Yuxiang Yang, Junjie Yang, Yufei Xu, Jing Zhang, Long Lan, and Dacheng Tao. Apt-36k: A large-scale benchmark for animal pose estimation and tracking. *Advances in Neural Information Processing Systems*, 35:17301–17313, 2022. 1, 3

[33] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021. 1, 2, 3

[34] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018. 3

[35] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5368, 2019. 1