Interspecies Knowledge Transfer for Facial Keypoint Detection

Maheen Rashid
University of California, Davis
mhnrashid@ucdavis.edu

Xiuye Gu*
Zhejiang University
gxy0922@zju.edu.cn

Yong Jae Lee
University of California, Davis
yongjaelee@ucdavis.edu

Abstract

We present a method for localizing facial keypoints on animals by transferring knowledge gained from human faces. Instead of directly finetuning a network trained to detect keypoints on human faces to animal faces (which is sub-optimal since human and animal faces can look quite different), we propose to first adapt the animal images to the pre-trained human detection network by correcting for the differences in animal and human face shape. We first find the nearest human neighbors for each animal image using an unsupervised shape matching method. We use these matches to train a thin plate spline warping network to warp each animal face to look more human-like. The warping network is then jointly finetuned with a pre-trained human facial keypoint detection network using an animal dataset. We demonstrate state-of-the-art results on both horse and sheep facial keypoint detection, and significant improvement over simple finetuning, especially when training data is scarce. Additionally, we present a new dataset with 3717 images with horse face and facial keypoint annotations.

1. Introduction

Facial keypoint detection is a necessary precondition for face alignment and registration, and impacts facial expression analysis, facial tracking, as well as graphics methods that manipulate or transform faces. While human facial keypoint detection is a mature area of research, despite its importance, animal facial keypoint detection is a relatively unexplored area. For example, veterinary research has shown that horses [16, 11], mice [25], sheep [3], and cats [17] display facial expressions of pain – a facial keypoint detector could be used to help automate such animal pain detection. In this paper, we tackle the problem of facial keypoint detection for animals, with a focus on horses and sheep.

Convolutional neural networks (CNNs) have demonstrated impressive performance for *human* facial keypoint detection [33, 47, 41, 54, 20, 61, 6, 56], which makes CNNs an attractive choice for learning facial keypoints on animals. Unfortunately, training a CNN from scratch typically requires large amounts of labeled data, which can be time-consuming and expensive to collect. Furthermore, while a CNN can be finetuned when there is not enough training data for the target task, a pre-trained network's extent of learning is limited both by the amount of data available for fine-tuning, as well as the *relatedness of the two tasks*. For example, previous work demonstrate that a network trained on man-made objects has limited ability to adapt to natural objects [52], and additional pretraining data is only beneficial when related to the target task [18].

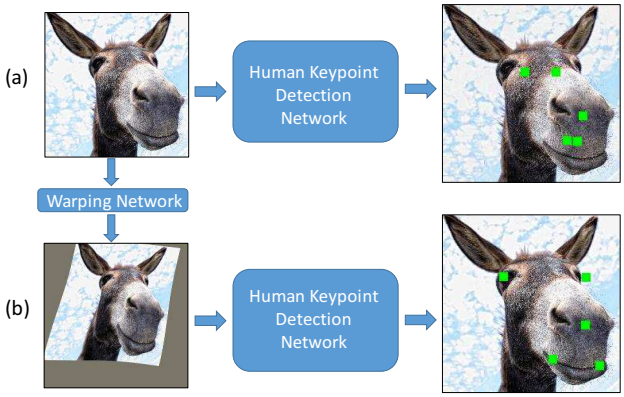

Figure 1. **Main idea.** (**a**) Directly finetuning a human keypoint detector to horses can be suboptimal, since horses and humans have very different shapes and appearances. (**b**) By warping a horse to have a more human-like shape, the pre-trained human keypoint detector can more easily adapt to the horse's appearance.

While there are large datasets with human facial keypoint annotations (e.g., AFLW has ∼26000 images [23]), there are, unfortunately, no large datasets of animal facial keypoints that could be used to train a CNN from scratch (e.g., the sheep dataset from [51] has only ∼600 images). At the same time, the structural differences between a human face and an animal face means that directly fine-tuning a human keypoint detector to animals can lead to a sub-optimal solution (as we demonstrate in Sec. 4).

In this paper, we address the problem of transferring knowledge between two different types of data (human and

*Work done while an intern at UC Davis.

animal faces) for the same task (keypoint detection). How can we achieve this with a CNN? Our key insight is that rather than adapt a pre-trained network to training data in a new domain, we can first do the *opposite*. That is, *we can adapt the training data from the new domain to the pre-trained network*, so that it is better conditioned for finetuning. By mapping the new data to a distribution that better aligns with the data from the pre-trained task, we can take a pre-trained network from the loosely-related task of human facial keypoint detection and finetune it for animal facial keypoint detection. Specifically, our idea is to explicitly warp each animal image to look more human-like, and then use the resulting warped images to finetune a network pre-trained to detect human facial keypoints. See Fig. 1.

Intuitively, by warping animal faces to look more human-like we can correct for their shape differences, so that during finetuning the network need only adapt to their differences in appearance. For example, the distance between the corners of a horse's mouth is typically much smaller than the distance between its eyes, whereas for a human these distances are roughly similar – a shape difference. In addition, horses have fur, and humans do not – an appearance difference. Our warping network adjusts for the shape difference by stretching out the horse's mouth corners, while during finetuning the keypoint detection network learns to adjust for the appearance difference.

Contributions. Our contributions are three fold: First, we introduce a novel approach for animal facial keypoint detection that transfers knowledge from the loosely-related domain of human facial keypoint detection. Second, we provide a new annotated horse facial keypoint dataset consisting of 3717 images. Third, we demonstrate state-of-the-art results on keypoint detection for horses and sheep. By transforming the animal data to look more human-like, we attain significant gains in keypoint detection accuracy over simple finetuning. Importantly, the gap between our approach and simple finetuning widens as the amount of training data is reduced, which shows the practical applicability of our approach to small datasets. Our data and code are available at https://github.com/menoRashid/animal_human_kp.

2. Related work

Facial landmark detection and alignment are mature topics of research in computer vision. Classic approaches include Active Appearance Models [8, 32, 35, 43], Constrained Local Models [10, 9, 36, 1], regression based methods [44, 48, 5, 49] with a cascade [13, 26, 59], and an ensemble of exemplar based models [2]. Recent work extends cascaded regression models by learning predictions from multiple domain-specific regressors [60] or by using a mixture of regression experts at each cascade level [42]. These models also demonstrate good performance when solved simultaneously with a closely related task, such as face detection [28], 3D face reconstruction [7], and facial action unit activation detection [46].

In the deep learning domain, coarse-to-fine approaches refine a coarse estimate of keypoints through a cascade [40, 58, 55, 56] or with branched networks [27]. Others assist keypoint detection by using separate cluster specific networks [45], augmenting it with related auxiliary tasks [57], initializing with head pose predictions [50], correcting for deformations with a spatial transformer [6], incorporating shape basis and thin plate spline transformations [53], formulating keypoint detection as a dense 3D face model fitting problem [20, 61], or using deep regression models in combination with de-corrupt autoencoders [54]. Recent work explore using recurrent neural networks [33, 47, 41].

While deep learning approaches demonstrate impressive performance, they typically require large annotated datasets. Rather than collect a large dataset, [31] uses domain specific augmentation techniques to *synthesize* pose, shape, and expression variations. However, it relies on the availability of 3D face models, and addresses the related but separate problem of face recognition. Similarly, [12] leverages large datasets available for face recognition to train a deep network, which is then used to guide training of an expression recognition network using only a small amount of data. However, while [12] transfers knowledge between two different tasks (face recognition and expression recognition) that rely on the same type of data (human faces), we transfer knowledge between two different data sources (human and animal faces) in order to solve the same task (facial keypoint detection).

To the best of our knowledge, *facial* keypoint detection in animals is a relatively unexplored problem. Very recently, [51] proposed an algorithm for keypoint detection in sheep, using triplet interpolated features in a cascaded shape regression framework. Unlike our approach, it relies on hand-crafted features and does not transfer knowledge from human to animal faces. Keypoint localization on birds has been explored in [39, 37, 30, 29], though these approaches do not focus on facial keypoint detection.

3. Approach

Our goal is to detect facial keypoints in animals without the aid of a large annotated animal dataset. To this end, we propose to adapt a pre-trained *human* facial keypoint detector to *animals* while accounting for their interspecies domain differences. For training, we assume access to keypoint annotated animal faces, and keypoint annotated human faces and their corresponding pre-trained human keypoint detector. For testing, we assume access to an animal face detector (i.e., we focus only on facial keypoint detection and not face detection).

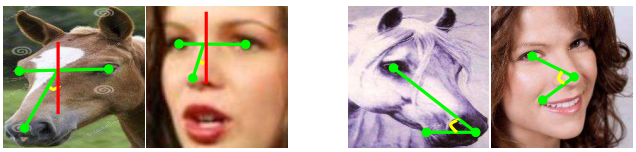

Figure 2. We approximate facial pose using the angle generated from the keypoint annotations. The keypoints used to compute the angle-of-interest depend on which facial parts are visible. For example, on the right, the horse's right eye and right mouth corner are not visible, so the three keypoints used are the left eye, nose, and left mouth corner. While simple, we find this approach to produce reliable pose estimates.

Our approach has three main steps: (1) finding nearest neighbor human faces that have similar pose to each animal face; (2) using the nearest neighbors to train an animal-to-human warping network; and (3) using the warped (human-like) animal images to fine-tune a pre-trained human keypoint detector for animal facial keypoint detection.

3.1. Nearest neighbors with pose matching

In order to fine-tune a (loosely-related) human facial keypoint detector to animals, our idea is to first warp the animal faces to have a more human-like shape so that it will be easier for the pre-trained human detector to adapt to the animal data. One challenge is that an arbitrary animal and human face pair can exhibit drastically different poses (e.g., a right-facing horse and a left-facing person), which can making warping extremely challenging or even impossible. To alleviate this difficulty, we first find animals and humans that are in similar poses.

If we had pose classifiers/annotations for both animal and human faces, then we could simply use their classifications/annotations to find compatible animal and human pairs. However, in this work, we assume we do not have access to pose classifiers nor pose annotations. Instead, we *approximate* a face pose given its keypoint annotations. More specifically, we compute the angular difference between a pair of human and animal keypoints, and then pick the nearest human faces for each animal instance.

For each animal training instance A_i, we find its nearest human neighbor training instance H_{j*} based on pose:

$$nn(A_i) = H_{j*} = \underset{H_j}{\operatorname{argmin}} |\measuredangle^* A_i - \measuredangle^* H_j|, \qquad (1)$$

where j indexes the entire human face training dataset, and the angle of interest $\measuredangle^*$ is measured in two different ways depending on the animal face's visible keypoints. When both eyes and the nose are present, we use $\measuredangle^* = \measuredangle N E_c V$, where E_c is the midpoint between the eye centers, N is the nose position, and V is a vertical line centered at E_c. If only the left eye is visible, then we use the left eye, nose, and left mouth keypoints: $\measuredangle^* = \measuredangle E_l N M_l$ (and $\measuredangle E_r N M_r$ if the right eye is visible). These cases are illustrated in Fig. 2.

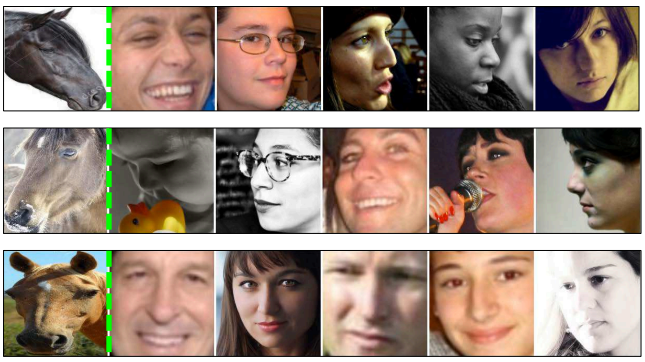

Figure 3. For each animal image (1st column), we find the nearest human neighbors in terms of pose. These human neighbors are used to train a warp network that warps an animal to have human-like face shape.

While simple, we find this approach to produce reliable pose estimates. In our experiments, we find the $K = 5$ nearest human neighbors for each animal face. Fig. 3 shows some examples. Since we use the TPS transformation for warping animals to humans (as described in the next section), we only compute matches for animal faces with at least three keypoints and ignore human matches whose keypoints are close to colinear, which can cause gross artifacts in warping. Note that we do not do pose matching during testing, since we do not have access to ground-truth keypoints; instead we rely on the ensuing warping network to have learned the "right" warp for each animal face pose (based on its appearance) during training.

3.2. Interspecies face warping network

Now that we have the nearest human faces (in terms of pose) for each animal face, we can use these matches to train an animal-to-human face warping network. This warping network serves to adapt the shape of the animal faces to more closely resemble that of humans, so that a pre-trained human facial keypoint detector can be more easily fine-tuned on animal faces.

For this, we train a CNN that takes as input an animal image and warps it via a thin plate spline (TPS) [4] transformation. Our warping network is a spatial transformer [19], with the key difference being that our warps are directly supervised, similar to [6].[1] Our network architecture is similar to the localization network in [38]; it is identical to Alexnet [24] up to the fifth convolutional layer, followed by a 1×1 convolution layer that halves the number of filters, two fully-connected layers, and batch normalization before every layer after the fifth. During training, the first five layers are pre-trained on ImageNet. We find these layer/filter

[1] In contrast, in [19] the supervision only comes from the final recognition objective e.g., keypoint detection. We show in Sec. 4 that direct warping supervision produces superior performance.

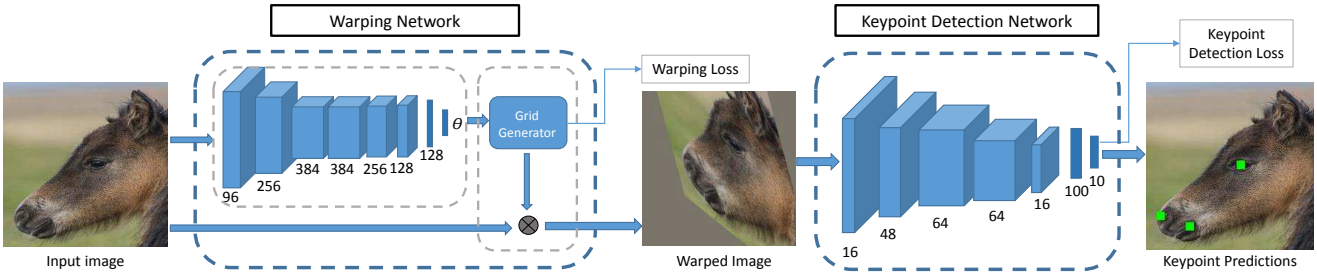

Figure 4. Our network architecture for animal facial keypoint detection. During training, the input image is fed into the warping network, which is directly supervised using keypoint-annotated human and animal image pairs with similar pose. The warping network warps the input animal image to have a human-like shape. The warped animal face is then passed onto the keypoint detection network, which finetunes a pre-trained human keypoint detection network with the warped animal images. During testing, the network takes the input image and produces 5 keypoint predictions for left eye, right eye, nose, left mouth corner, and right mouth corner.

choices to enable good TPS transformation learning without overfitting. See Fig. 4 (left).

For each animal and human training image pair, we first calculate the ground-truth TPS transformation using its corresponding keypoint pairs and apply the transformation to produce a ground-truth warped animal image. We then use our warping network to compute a predicted warped animal image. To train the network, we regress on the difference between the ground-truth warped image and predicted warped image pixel position offsets, similar to [21]. Specifically, we use the squared loss to train the network:

$$L_{warp}(A_i) = \sum_m (p_{i,m}^{pred} - p_{i,m}^{gt})^2, \quad (2)$$

where A_i is the i-th animal image, $p_{i,m}^{pred}$ and $p_{i,m}^{gt}$ are the predicted offset and ground-truth offset, respectively, for pixel m.

It is important to note that our warping network requires no additional annotation for training, since we only use the animal/human keypoint annotations to find matches (which are already available and necessary for training their respective keypoint detectors). In addition, since each animal instance has multiple ($K = 5$) human matches, the warping network is trained to identify multiple transformations as potentially correct. This serves as a form of data augmentation, and helps make the network less sensitive to outlier matches.

3.3. Animal keypoint detection network

Our warping network from the previous section conditions the distribution of the animal data to more closely resemble human data, so that we can harness the large *human* keypoint annotated datasets that are readily available for *animal* keypoint detection. The final step is to finetune a pre-trained human facial keypoint detection network to detect facial keypoints on our warped animal faces.

Our keypoint detector is a variant of the Vanilla CNN architecture used in [45]. The network has four convolutional layers, and two fully-connected layers with absolute $\tanh$ non-linearity, and max-pooling in the last three convolutional layers. We adapt it to work for larger images—we use 224×224 images as input rather than 40×40 used in [45]—by adding an extra convolutional and max-pooling layer. In addition, we add batch normalization after every layer since we find the $\tanh$ layers in the original network to be prone to saturation. Fig. 4 (right) shows the architecture. Our keypoint detection network is pre-trained on human facial keypoints on the AFLW [23] dataset and the training data used in [40] (a total of 31524 images).

To finetune our keypoint network, we use the smooth $L1$ loss (equivalent to the Huber loss with δ=1) used in [15] since it is less sensitive to outliers that may occur with unusual animal poses:

$$L_{keypoint}(A_i) = \sum_n smooth_{L_1}(k_{i,n}^{pred} - k_{i,n}^{gt}), \quad (3)$$

where A_i is the i-th animal image, $k_{i,n}^{pred}$ and $k_{i,n}^{gt}$ are the predicted and ground-truth keypoint position, respectively, for the n-th keypoint, and $smooth_{L_1}$ is

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \quad (4)$$

We set the loss for predicted keypoints with no corresponding ground-truth annotation (due to occlusion) to zero.

3.4. Final architecture

In our final model, we fit the warping network before a keypoint detection network that is pre-trained on human keypoint detection. We use the two losses to jointly finetune both networks. The keypoint detection loss $L_{keypoint}$ (Eqn. 3) is back propagated through both the keypoint detection network, as well as the warping network. Additionally, the warping loss L_{warp} (Eqn. 2) is backpropagated

through the warping network, and the gradients are accumulated before the weights for both networks are updated. See Fig. 4.

In the testing phase, our keypoint network predicts all 5 facial keypoints for every image. In our experiments, we do not penalize the network for keypoint predictions that are not visible in the image and results are reported only for predicted keypoints that have corresponding ground-truth annotation. For evaluation, the keypoints predicted on warped images are transferred back to the original image using the TPS warp parameters.

3.5. Horse Facial Keypoint dataset

As part of this work, we created a new horse dataset to train and evaluate facial keypoint detection algorithms. We collected images through Google and Flickr by querying for "horse face", "horse head", and "horse". In addition, we included images from the PASCAL VOC 2012 [14] and Imagenet 2012 [34] datasets. There are a total of 3717 images in the dataset: 3531 for training, and 186 for testing. We annotated each image with face bounding boxes, and 5 keypoints: left eye center, right eye center, nose, left mouth corner, and right mouth corner.

4. Experiments

In this section, we analyze our model's keypoint detection accuracy, and perform ablation studies to measure the contribution of each component. In addition, we evaluate our method's performance as the amount of training data is varied, and also measure an upper-bound performance if animal-to-human warping were perfect.

Baselines. We compare against the algorithm presented in [51], which uses triplet-interpolated features (TIF) in a cascaded shape regression framework for keypoint detection on animals. We also develop our own baselines. The first baseline is our full model without the warping network. It simply finetunes the pre-trained human facial keypoint network on the animal dataset ("BL FT"). The second baseline is our full model without the warping loss; i.e., it finetunes the pre-trained human facial keypoint network and the warping network with only the keypoint detection loss. This baseline is equivalent to the spatial transformer setting presented in [19]. We show results for this with TPS warps ("BL TPS"). The third baseline trains the keypoint detection network from scratch; i.e., without any human facial keypoint detection pretraining and without the warping network ("Scratch").

Datasets. We pretrain our keypoint detection network on human facial keypoints from the AFLW [23] dataset and the training data used in [40] (a total of 31524 images). This dataset is also used for animal to human nearest neighbor retrieval. We evaluate keypoint detection on two animals: horses and sheep. For the horse experiments, we use our Horse Facial Keypoint dataset, which consists of 3531 images for training and 186 for testing. For the sheep experiments, we manually annotated a subset of the dataset provided in [51] with mouth corners so that we have the same 5 keypoints present in the human dataset. The dataset consists of 432 images for training and 99 for testing.

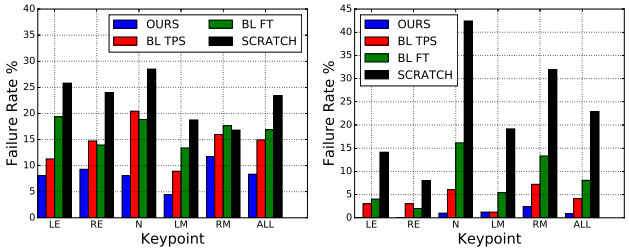

Figure 5. Average keypoint detection failure rate (% of predicted keypoints whose euclidean distance to the corresponding ground-truth keypoint is more than 10% of the face bounding box size). Horses (**left**) and Sheep (**right**). Our approach outperforms the baselines. *Lower is better.* See text for details.

Evaluation metric. We use the same metric for evaluation as [51]: If the euclidean distance between the predicted and ground-truth keypoint is more than 10% of the face (bounding box) size, it is considered a failure. We then compute the average failure rate as the percentage of testing keypoints that are failures.

Training and implementation details. We find that pretraining the warping network before joint training leads to better performance. To train the warping and keypoint network, we use $K = 5$ human neighbors for each animal instance. These matches are also used to supervise the "GT Warp" network described in Sec. 4.4.

For the TPS warping network, we use a 5×5 grid of control points. We optimize all networks using Adam [22]. The base learning rate for the warp network training is 0.001, with a $\frac{1}{10}\times$ lower learning rate for the pre-trained layers. It is trained for 50 epochs, with the learning rate lowered by $\frac{1}{10}\times$ after 25 epochs. During full system training, the warp network has the same learning rates, while the keypoint detection network has a learning rate of 0.01. We train the network for 150 epochs, lowering the learning rate twice after 50 and 100 epochs. Finally, we use horizontal flips and rotations from -10° to 10° at increments of 5° for data augmentation.

4.1. Comparison with our baselines

We first compare our full model with our model variant baselines. Figure 5 (left) and (right) show results on horse and sheep data, respectively. We outperform all of our baselines significantly for both horses and sheep, with an aver-

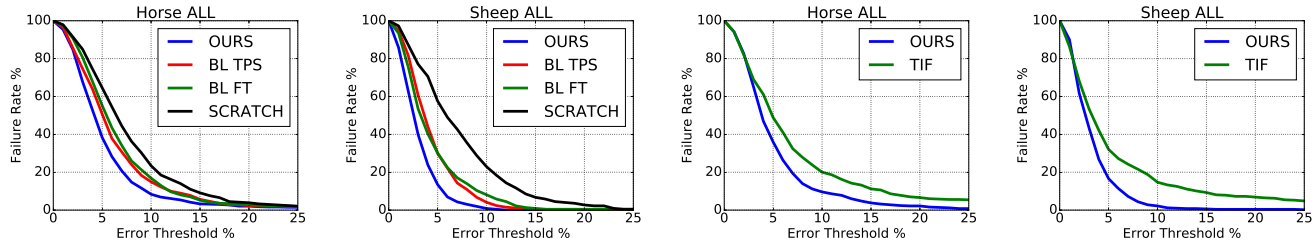

Figure 6. Average keypoint detection failure rate across all keypoints for our system vs. our baselines (first two plots) and the Triplet Interpolated Features (TIF) approach of Yang et al. [51] (last two plots). Our system sustains lower failure rates across stricter failure thresholds than all baselines.

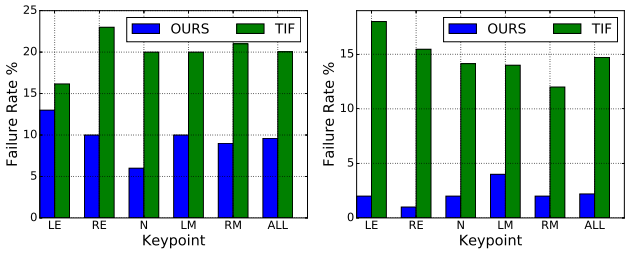

Figure 7. Average keypoint detection failure rate for Horses (**left**) and Sheep (**right**). Our approach significantly outperforms the Triplet Interpolated Features (TIF) approach of Yang et al. [51], which combines hand-crafted features with cascaded shape regressors. *Lower is better.*

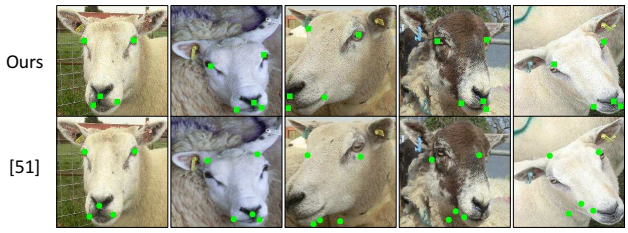

Figure 8. Qualitative examples comparing our approach and Yang et al. [51] on their Sheep dataset. While [51] can produce good predictions (first column), overall, our method produces significantly more accurate results.

age failure rate across keypoints at 8.36% and 0.87%, respectively.

Overall, the failure rate for all methods (except Scratch) for sheep is lower than that for horses. The main reason is due to the pose distribution of human and sheep data being more similar than that of human and horse data. The human and sheep data have 72% and 84% of images in frontal pose (faces with all 5 keypoints visible) as compared to only 29% for horses. The majority (60%) of horse faces are side-view (faces with only 3 keypoints visible). This similarity makes it easier for the human pre-trained network to adapt to sheep than to horses. Nonetheless, the fact that our method outperforms the baselines for both datasets, demonstrates that our idea is generalizable across different types of data.

These results also show the importance of each component of our system. Training with a human pre-trained network does better than training from scratch (BL FT vs. Scratch); adding a warping network that is only *weakly-guided* by the keypoint detection loss further improves results (BL TPS vs. BL FT); and finally, directly supervising the warping network to produce animal faces that look more human-like leads to the best performance (Ours vs. BL TPS). The first two plots in Fig. 6 show the results of varying the acceptance threshold (on the euclidean distance between the ground-truth and predicted keypoint) for a valid keypoint on our and the baselines' performance. Our method sustains superior accuracy across thresholds, which again indicates that we predict keypoints more accurately.

Fig. 9 shows qualitative examples of predicted keypoints and predicted warps for ours and the baselines. Noticeably, the TPS warps produced without the warping loss (BL TPS Warp) fail to distinguish between the different horse poses, and also do not warp the horse faces to look more human like. On the other hand, our warping network is able to do both tasks well since it is directly supervised by pose specific human matches. By warping the horses to have more human-like shape, our method produces more precise keypoint predictions than the baselines. The last two rows show typical failure examples due to extreme pose or occlusion.

4.2. Comparison with Yang et al. [51]

We next compare our method to the Triplet Interpolated Features (TIF) approach of [51], which is the state-of-the-art animal keypoint detector. The method requires the existence of all landmarks in all training examples. We therefore picked a subset of the horse and sheep images where all 5 keypoints are visible and marked: 345/100 train/test images for sheep, and 982/100 train/test images for horses.

Fig. 8 shows qualitative examples comparing our method's keypoint predictions vs. those made by TIF. TIF often fails to handle large appearance and pose variations. This is also reflected in the quantitative results, which are shown in Fig. 6 (third) and Fig. 7 (left) for the horse dataset and Fig. 6 (fourth) and Fig. 7 (right) for the sheep dataset. We significantly outperform TIF on both datasets (10.44% and 12.52% points lower failure rate for horses and sheep,

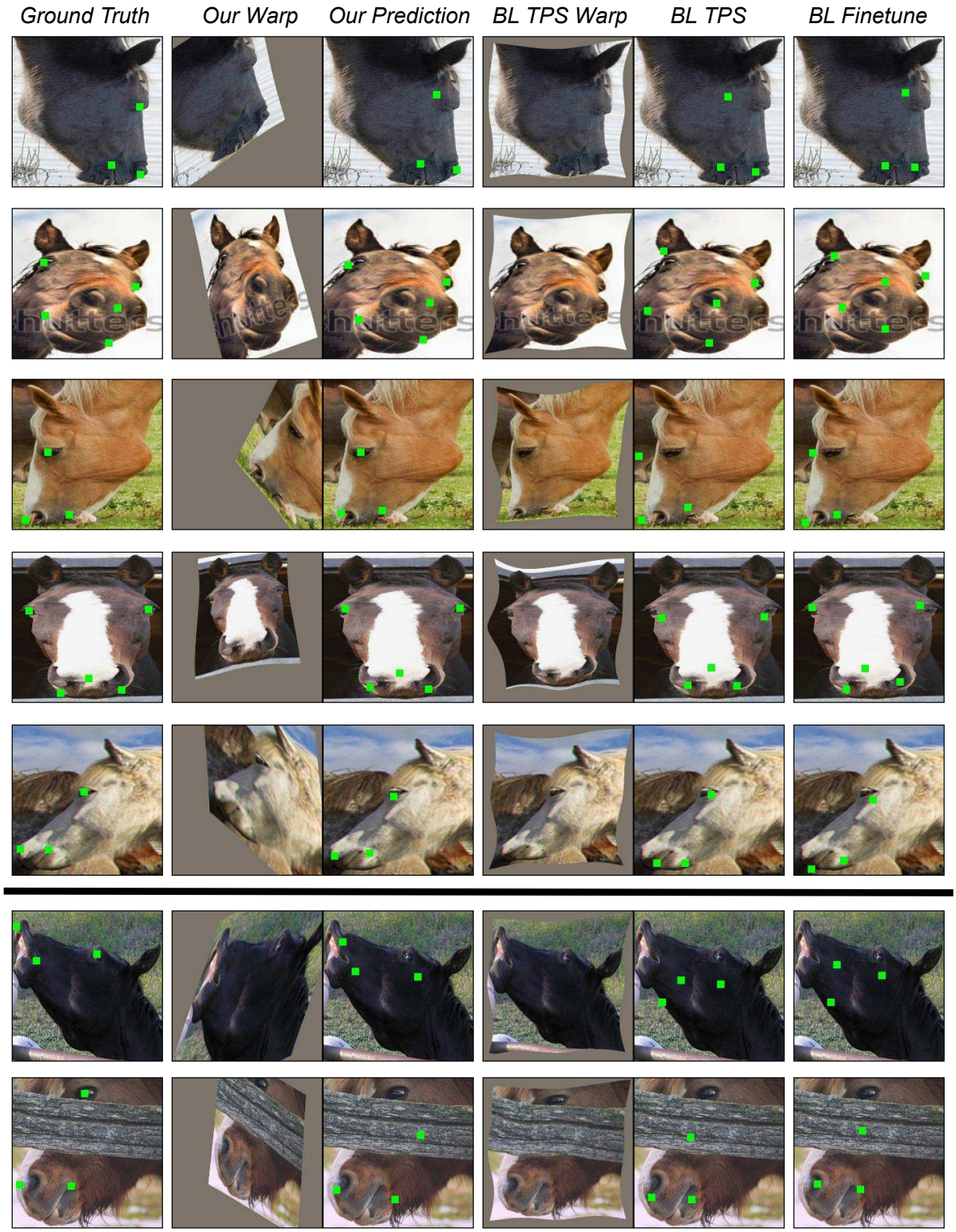

Figure 9. Qualitative examples of predicted keypoints and predicted warps for ours and the baselines. The first five rows show examples where our method outperforms the baseline. While the baselines also produce reasonable results, by warping the horses to have more human-like shape, our method produces more precise keypoint predictions. For example, in the first row, the baselines do not localize the nose and mouth corner as well as ours. The last two rows show typical failure examples due to extreme pose or occlusion.

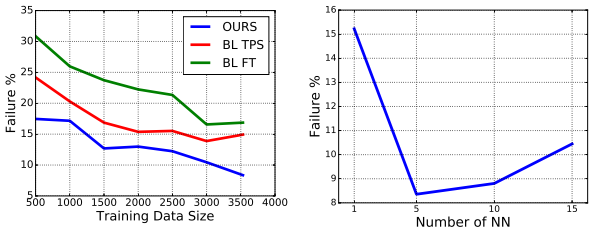

Figure 10. (**left**) Average keypoint detection failure rate as a function of the number of training instances on the Horse dataset. Our failure rate increases more gracefully compared to the baselines as the number of training images is decreased. *Lower is better.* (**right**) Increasing the number of human face neighbors for an animal face instance increases performance until noisy neighbors cause performance to drop.

respectively). The main reason is because we use a high capacity deep network, whereas TIF is a shallow method that learns with hand-crafted features. Importantly, the reason that we are able to use such a high capacity deep network—despite the limited training data of the animal datasets—is precisely because we correct for the shape differences between animals and humans in order to *finetune* a pre-trained human keypoint detection network.

4.3. Effect of training data size

In this section, we evaluate how the performance of our network changes as the amount of training data varies. For this, we train and test multiple versions of our model and the baselines, each time using 500 to 3531 training images in 500 image increments on the Horse dataset.

Figure 10 (left) shows the result. While the performance of all methods decreases with the training data amount, our performance suffers much less than that of the simple finetuning and TPS baselines. In particular, when using only 500 training images, our method has a 6.72% point lower failure rate than the TPS baseline while relying on the same network architecture, and a 13.39% point lower failure rate than simple finetuning, *without using any additional training data or annotations*.

This result demonstrates that our algorithm adapts well to small amounts of training data, and bolsters our original argument that explicitly correcting for interspecies shape differences enables better finetuning, since the pre-trained human keypoint detection network can mostly focus on the appearance differences between the two domains (humans and animals). Importantly, it also shows the practical applicability of our approach to small datasets.

4.4. Effect of warping accuracy

We next analyze the influence of warping accuracy on keypoint detection. For this, we first analyze the performance of our keypoint detection network when finetuned with *ground-truth* warped images ("GT Warp"), where we use the ground-truth keypoint annotations between human and horse faces for warping (i.e., the keypoint detection network is finetuned with ground-truth warped images). In a sense, this represents the upper bound of the performance of our system.

	GT Warp	Ours
Failure Rate %	7.76%	8.36%

Table 1. Average keypoint detection failure rate across all keypoints on the Horse dataset, comparing our approach to an upper-bound ground-truth warping baseline. *Lower is better.*

Table 1 shows the results on our Horse dataset. First, the GT Warp upper-bound produces even lower error rates than our method, which demonstrates the efficacy of the idea of correcting for shape differences by warping. At the same time, the non-negligible error rate of GT Warp also hints at the limitation of our warping network's training data and/or pose matching strategy. Better training data, with either a different algorithm for nearest pose neighbor matching or an increase in the keypoints that are annotated could potentially lead to a better upper-bound, and would likely provide improvements for our approach as well.

4.5. Evaluation of Nearest Neighbors

Finally, we evaluate the importance of human nearest neighbors for our system. We vary the number of nearest neighbors used for training our full system from $K = 1$ to $K = 15$ at increments of 5 for our full Horse training set. The result is shown in Figure 10 (right). While the error rate decreases as the number of neighbors used for training is increased in the beginning, eventually, the noise in retrieved nearest neighbors causes the error rate to increase.

5. Conclusion

We presented a novel approach for localizing facial keypoints on animals. Modern deep learning methods typically require large annotated datasets, but collecting such datasets is a time consuming and expensive process.

Rather than collect a large annotated animal dataset, we instead warp an animal's face shape to look like that of a human. In this way, our approach can harness the readily-available human facial keypoint annotated datasets for the loosely-related task of animal facial keypoint detection. We compared our approach with several strong baselines, and demonstrated state-of-the-art results on horse and sheep facial keypoint detection. Finally, we introduced a novel Horse Facial Keypoint dataset, which we hope the community will use for further research on this relatively unexplored topic of animal facial keypoint detection.

Acknowledgements. This work was supported in part by a gift from the Swedish University of Agricultural Sciences and GPUs donated by NVIDIA.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013. 2

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *PAMI*, 35(12):2930–2940, 2013. 2

[3] A. Boissy, A. Aubert, L. Désiré, L. Greiveldinger, E. Delval, and I. Veissier. Cognitive sciences to relate ear postures to emotions in sheep. *Animal Welfare*, 20(1):47, 2011. 1

[4] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *TPAMI*, 11(6):567–585, 1989. 3

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *IJCV*, 107(2):177–190, 2014. 2

[6] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In *ECCV*, 2016. 1, 2, 3

[7] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, 2014. 2

[8] T. F. Cootes, G. J. Edwards, C. J. Taylor, et al. Active appearance models. *TPAMI*, 23(6):681–685, 2001. 2

[9] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008. 2

[10] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006. 2

[11] E. Dalla Costa, M. Minero, D. Lebelt, D. Stucke, E. Canali, and M. C. Leach. Development of the Horse Grimace Scale (HGS) as a pain assessment tool in horses undergoing routine castration. *PLoS one*, 9(3):e92281, 2014. 1

[12] H. Ding, S. K. Zhou, and R. Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. *arXiv preprint arXiv:1609.06591*, 2016. 2

[13] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010. 2

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 5

[15] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 4

[16] K. B. Gleerup, B. Forkman, C. Lindegaard, and P. H. Andersen. An equine pain face. *Veterinary anesthesia and analgesia*, 42(1):103–114, 2015. 1

[17] E. Holden, G. Calvo, M. Collins, A. Bell, J. Reid, E. Scott, and A. Nolan. Evaluation of facial expression in acute pain in cats. *Journal of Small Animal Practice*, 55(12):615–621, 2014. 1

[18] M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 1

[19] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 3, 5

[20] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3D model fitting. In *CVPR*, 2016. 1, 2

[21] A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. *CVPR*, 2016. 4

[22] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *BeFIT Workshop*, 2011. 1, 4, 5

[24] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 3

[25] D. J. Langford, A. L. Bailey, M. L. Chanda, S. E. Clarke, T. E. Drummond, S. Echols, S. Glick, J. Ingrao, T. Klassen-Ross, M. L. LaCroix-Fralish, et al. Coding of facial expressions of pain in the laboratory mouse. *Nature methods*, 7(6):447–449, 2010. 1

[26] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *CVPR*, 2015. 2

[27] Z. Liang, S. Ding, and L. Lin. Unconstrained facial landmark localization with backbone-branches fully-convolutional networks. *arXiv preprint arXiv:1507.03409*, 2015. 2

[28] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3d face reconstruction. In *ECCV*, 2016. 2

[29] J. Liu and P. N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *ICCV*, 2013. 2

[30] J. Liu, Y. Li, and P. N. Belhumeur. Part-pair representation for part localization. In *ECCV*, 2014. 2

[31] I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? *arXiv preprint arXiv:1603.07057*, 2016. 2

[32] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 2

[33] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *ECCV*, 2016. 1, 2

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 5

[35] J. Saragih and R. Goecke. A nonlinear discriminative approach to aam fitting. In *ICCV*, 2007. 2

[36] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 2

[37] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem. Part localization using multi-proposal consensus for fine-grained categorization. *BMVC*, 2015. 2

[38] K. K. Singh and Y. J. Lee. End-to-end localization and ranking for relative attributes. In *ECCV*, 2016. 3

[39] S. Singh, D. Hoiem, and D. Forsyth. Learning to localize little landmarks. In *CVPR*, 2016. 2

[40] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013. 2, 4, 5

[41] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, 2016. 1, 2

[42] O. Tuzel, S. Tambe, and T. K. Marks. Robust face alignment using a mixture of invariant experts. In *ECCV*, 2016. 2

[43] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *CVPR*, 2013. 2

[44] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, 2010. 2

[45] Y. Wu and T. Hassner. Facial landmark detection with tweaked convolutional neural networks. *arXiv preprint arXiv:1511.04031*, 2015. 2, 4

[46] Y. Wu and Q. Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *CVPR*, 2016. 2

[47] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *ECCV*, 2016. 1, 2

[48] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 2

[49] X. Xiong and F. De la Torre. Global supervised descent method. In *CVPR*, 2015. 2

[50] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson. Face alignment assisted by head pose estimation. *arXiv preprint arXiv:1507.03148*, 2015. 2

[51] H. Yang, R. Zhang, and P. Robinson. Human and sheep facial landmarks localisation by triplet interpolated features. In *WACV*, 2016. 1, 2, 5, 6

[52] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 1

[53] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *ECCV*, 2016. 2

[54] J. Zhang, M. Kan, S. Shan, and X. Chen. Occlusion-free face alignment: Deep regression networks coupled with de-corrupt autoencoders. In *CVPR*, 2016. 1, 2

[55] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, 2014. 2

[56] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *arXiv preprint arXiv:1604.02878*, 2016. 1, 2

[57] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 2

[58] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCV Workshops*, 2013. 2

[59] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015. 2

[60] S. Zhu, C. Li, C. C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *CVPR*, 2016. 2

[61] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *CVPR*, 2016. 1, 2