Domain Adaptation for Enhancing Visual Hand Landmark Prediction AI in Infrared Imaging

Application in Early Diagnosis of Leprosy

Vladimir Sachkov[1]

Responsible Professor: Jan van Gemert[1]
Supervisors: Zhi-Yi Lin[1]**, Thomas Markhorst**[1]
Examiner: Kaitai Liang[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty
Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 26, 2025

Name of the student: Vladimir Sachkov
Final project course: CSE3000 Research Project
Thesis committee: Jan van Gemert, Zhi-Yi Lin, Thomas Markhorst, Kaitai Liang

Abstract

In this work, we investigate how domain adaptation techniques can improve the performance of hand landmark detection models originally trained on RGB images when deployed on infrared (IR) data. Our motivation stems from a medical use case in Nepal, where clinicians require reliable temperature estimation at hand keypoints to detect early signs of leprosy. We evaluate three methods on a small IR dataset (80 labeled images & 5000 unlabeled frames): a shallow adaptation (AdaBN), a deep alignment approach (Deep CORAL), and a test-time subspace alignment method (SSA).

Our experiments show that while AdaBN and SSA yield moderate improvements, Deep CORAL achieves stronger gains through targeted training of specific model components. The combination of these methods produces superior results, yielding an 11% improvement in percentage of correct keypoints (PCK@0.05) on our custom annotated IR dataset.

These findings demonstrate that combining lightweight and deep domain adaptation approaches can effectively enhance IR hand landmark detection accuracy without requiring large labeled datasets, enabling practical deployment for clinical thermal imaging in resource-limited settings.

Keywords: domain adaptation, infrared imaging, test-time adaptation, medical imaging, hand landmark detection, AdaBN, Deep CORAL, subspace alignment.

1 Introduction

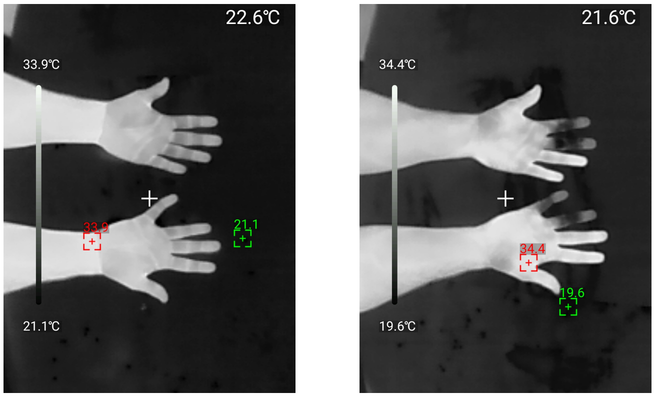
Motivation. Infrared (IR) imaging is crucial in certain medical scenarios where measuring temperature from hand keypoints is required. For example, leprosy detection relies on accurate temperature measurements at anatomical joints in the hand, which are not obtainable from standard RGB images [17]. Although multiple hand landmark detection methods can accurately predict 21 skeletal keypoints (as in Figure 2) from RGB images, no model to date is designed to handle IR images of hands. Collecting a large dataset of annotated IR hand images is challenging and expensive, motivating us to investigate whether existing RGB-trained models can be adapted to the IR domain through *domain adaptation*.

Domain Adaptation. Domain adaptation is a subfield of machine learning and transfer learning that addresses the challenge of training a model on one data distribution (the source domain) and applying it to a different but related distribution (the target domain) [20]. In *unsupervised domain adaptation*, the target-domain data have no annotations, which suits scenarios where labeled IR images are scarce. Further, *test-time adaptation* explores adjusting the model at inference stage, even when the source (RGB) dataset is not accessible during testing.

Problem Statement. Although the MediaPipe Hands model [22] and its related versions (e.g., BlazePalm, BlazeHand [24]) utilized in related IR hand landmark detection research conducted by Schemkes [17] have shown high accuracy on RGB data, they are difficult to retrain effectively for new domains due to limited source code and minimal publicly available training layers. These models detect 21 skeletal keypoints in each hand (where the points correspond to anatomical joints), but direct application to IR images significantly drops in accuracy. Moreover, our dataset of IR-labeled images is extremely small (only 80 labeled images, plus 5000 unlabeled images), making standard supervised retraining infeasible. This work thus seeks to determine whether unsupervised and test-time domain adaptation methods can improve IR hand landmark detection without requiring a large, annotated IR dataset.

Approach Overview. We focus on adapting an existing RGB-based hand landmark detection pipeline for IR images. Specifically, we use Facebook's InterWild model [4], an open-source hand pose estimation framework that offers greater flexibility for adaptation compared to closed-source alternatives. We employ three unsupervised domain adaptation strategies and a test-time adaptation approach, avoiding the need for IR labels and sidestepping the full retraining of the source model. Because memory and efficiency are also concerns, we investigate *spatial reduction* (i.e., reducing input resolution while preserving essential spatial information) to mitigate computational overhead.

Example IR Images. Figure 1 shows IR examples in which the hands have different exposures to cold, resulting in partial occlusion and varying brightness. The left image exhibits a normal temperature distribution, while the right image displays colder fingertips:



(a) Infrared hand with no cold exposure.

(b) Infrared hand with partial cold exposure.

Figure 1: Example IR images under different temperature conditions and partial occlusions.

Research Question. Ultimately, this research aims to answer:

> *To what extent can domain adaptation techniques improve the performance of a hand landmark de-*

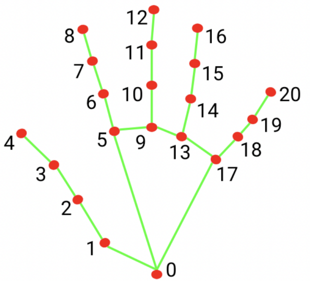


Figure 2: Hand keypoint structure showing the 21 skeletal points detected per hand.

tection model trained on RGB images when tested on IR images?

By investigating unsupervised and test-time adaptation, we test the hypothesis that such techniques can substantially enhance IR-based hand keypoint detection under small-labeled-data constraints. The findings could benefit medical practitioners by providing more robust and accurate IR-based hand landmark detection, aiding early diagnosis of leprosy and supporting other domains where IR imaging is critical.

Summary of Contributions. Our work makes several key contributions to domain adaptation for infrared hand landmark detection.

First, we construct and release a novel multimodal dataset containing aligned RGB-IR image pairs with systematic temperature variations (as seen in Figure 1), annotated with hand keypoint locations to enable cross-domain validation.

Our comprehensive evaluation framework introduces adaptive PCK metrics that account for hand pose variability, coupled with visualization tools that reveal both quantitative improvements and qualitative feature alignment patterns.

We implement and evaluate three distinct adaptation approaches (AdaBN [11], Deep CORAL [18], and SSA [7]) on the InterWild architecture, demonstrating that combined methods achieve an 11% improvement over baseline through complementary parameter space optimization on our custom small-scale IR dataset.

Through extensive experimentation, we observed that spatial reduction techniques like average adaptive pooling can maintain model accuracy, while reducing computational overhead, crucial for real-world deployment.

2 Literature Review

Domain adaptation research [5; 23] highlighted three viable approaches. AdaBN [11] offered simple batch normalization layer [6] statistic recalibration, compatible with InterWild's accessible layers. For deeper adaptation, Deep CORAL [18] provided covariance alignment through its CORAL loss, though its joint training requirement posed challenges with our 200GB source datasets. This made test-time adaptation methods appealing, though most classification-oriented approaches [5] proved unsuitable for our regression task.

Regression-specific methods revealed two candidates: SSA [7] for subspace alignment and RegDA [8] for heatmap adaptation. While RegDA showed theoretical promise, its adversarial framework added complexity compared to SSA's simpler implementation. Our constrained IR dataset (5000 frames) further favored SSA's efficient target-only training, aligning with the practical considerations outlined in Section 4.

The final selection—AdaBN for shallow adaptation, Deep CORAL for feature alignment, and SSA for test-time adaptation—balanced performance with our technical constraints: regression-focused outputs, limited target data, limited computational resources, and large unwieldy source domains.

3 Methodology

3.1 Model Architecture

Model Structure. The InterWild model [14] architecture was modified to focus solely on 2D skeleton prediction by removing the 3D mesh reconstruction layers. The streamlined architecture consists of four primary components operating in sequence:

The body_backbone network serves as the initial feature extractor, processing the input image to generate rich visual features. These features are then processed by body_box_net for hand bounding box detection. The detected regions, along with the original image, are passed to hand_roi_net, which extracts hand-specific features for both left and right hands independently. Finally, hand_detection_net processes these features to predict the final keypoint coordinates.

The model's forward pass can be configured to operate in different modes: full prediction (both detection and keypoints), detection-only using only_bbox flag, or keypoint-only prediction with only_hand flag when bounding boxes are pre-computed. This modular design enables targeted training and inference optimizations. The complete architecture and data flow are illustrated in Figure 3.

3.2 Domain Adaptation Methods

Domain Adaptation Techniques implementation.

- **AdaBN[11]:** A shallow method focusing on re-estimating batch normalization statistics using the target (IR) domain. The original paper describes the need to calculate mean and variance of neuron responses on all target domain samples for each batch normalization layer, specifically by concatenating all neuron responses right before BN layer $\mathbf{x}_j = [..., x_j(m), ...]$ from the target domain t and computing their statistics as $\mu_j^t = \mathrm{E}(\mathbf{x}_j^t)$ and $\sigma_j^t = \sqrt{\mathrm{Var}(\mathbf{x}_j^t)}$, and then those statistics should be replaced inside each BN layer. But the original paper provides no implementation details for modern deep learning frameworks.

  Working with PyTorch's BatchNorm2d [2] implementation, several key implementation considerations were identified. The BatchNorm layers can operate in either train or eval mode. In eval mode, pre-computed statistics from training are used, while train mode uses current batch statistics and updates running estimates according to $\hat{x}_{\text{new}} = (1 - \text{momentum}) \times \hat{x} + \text{momentum} \times x_t$, where $\hat{x}$ represents accumulated statistics and x_t is current batch statistics.

This understanding led to multiple possible implementation approaches: directly updating statistics using all target domain data at once, gradually updating through batches with momentum, or employing a mixed approach using different modes for different BN layers. The implementation required careful tuning of several hyperparameters, including the batch size for statistics computation, momentum value for running statistics updates, BN layer modes (train/eval) for different network components, and the dataset sampling strategy during inference. These considerations formed the hyperparameter search space for our AdaBN implementation.

- **Deep CORAL [18]:** A deep method incorporating a CORAL loss term to align features between source and target domains during training. We handle high feature dimensionality by applying spatial pooling [1] before computing the covariance-based CORAL loss. Specifically, we use adaptive average pooling, which reduces spatial dimensions while preserving feature characteristics by averaging values within dynamically-sized regions. For an input tensor X of size $C \times H \times W$, the output Y of size $C \times H' \times W'$ is computed as:

$$Y_{c,h',w'} = \frac{1}{|R_{h',w'}|} \sum_{(i,j)\in R_{h',w'}} X_{c,i,j} \tag{1}$$

where $R_{h',w'}$ represents the set of pixels in the input region that map to output position (h', w'), and $|R_{h',w'}|$ is the size of this region. This operation reduces our feature maps to $C \times 1 \times 1$ before computing the CORAL loss.

The CORAL loss is defined as:

$$\ell_{\text{CORAL}} = \frac{1}{4d^2} \|C_S - C_T\|_F^2 \tag{2}$$

where C_S and C_T are the feature covariance matrices from the source and target domains respectively, d is the feature dimension, and $\| \cdot \|_F^2$ denotes the squared Frobenius norm, as defined in [18].

The total loss for each module combines task-specific supervision losses with the CORAL loss:

$$\mathcal{L}_{\text{body}} = \mathcal{L}_{\text{bbox}} + \lambda \ell_{\text{CORAL}}^{\text{body}} \tag{3}$$

$$\mathcal{L}_{\text{hand}} = \mathcal{L}_{\text{kp}} + \lambda \ell_{\text{CORAL}}^{\text{hand}} \tag{4}$$

where $\mathcal{L}_{\text{bbox}}$ represents the bounding box detection losses, $\mathcal{L}_{\text{kp}}$ is the keypoint prediction loss (computed as absolute coordinate differences), λ is the CORAL loss weight, and $\ell_{\text{CORAL}}^{\text{body}}$ and $\ell_{\text{CORAL}}^{\text{hand}}$ are the CORAL losses computed at the body and hand feature levels respectively.

This loss was applied at two levels of the model, as shown in Figure 3. This is done since, model contains two different feature representations, one for detecting bounding boxes, and one for hand keypoints. To facilitate efficient training and feature alignment, we modified the model's forward method to process both target and source domain batches simultaneously, enabling CORAL loss computation at both feature levels. Coral_weight is a crucial hyperparameter for this method, since by increasing it too much we could achieve higher feature consistency across domain, but lose model's predictive power.

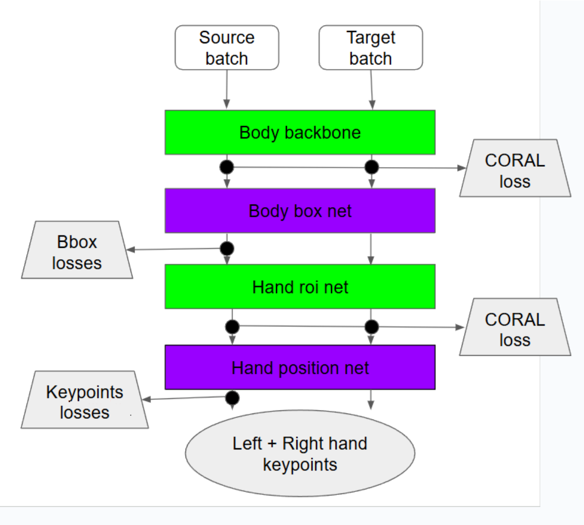


Figure 3: Schematic representation of Deep CORAL training process in the InterWild model. The model consists of two main modules: (1) hand detection module (left) with body_backbone (feature extractor, green) and body_box_net (box predictor, purple), and (2) keypoint prediction module (right) with hand_roi_net (feature extractor, green) and hand_detection_net (keypoint predictor, purple). CORAL loss aligns feature representations between source (RGB) and target (IR) domains at both feature extractor outputs, while supervision losses are computed from predictor outputs.

- **SSA (Test-time Adaptation for Regression by Subspace Alignment) [7]:** A test-time adaptation technique specifically designed for regression tasks that operates only on the target domain during training. SSA updates only the affine parameters (γ and β) of BN layers (only for feature extraction layers, body_backbone and hand_roi_net in our case), while keeping other weights frozen.

The method addresses a key challenge in regression models where features tend to be less diverse than in classification tasks, often distributed in a small subspace with many dimensions having zero variance. This makes naive feature alignment unstable due to variance terms in the denominator of KL divergence calculations.

SSA performs subspace detection by computing covariance matrices, and means of source domain features from body_backbone and hand_roi_net, calculating their eigenvectors and eigenvalues, projecting target domain features onto the space mapped by top-K eigenvectors, computing mean and variance of the projected vectors, and finally calculating the loss between two Gaussians in this subspace, where the source distribution has zero mean and eigenvalues as variance.

The projection of target domain features into the source subspace is given by:

$$\mathbf{z}_i^t = \mathbf{V}^s(\mathbf{z}_i^t - \boldsymbol{\mu}^s), \tag{5}$$

where $\mathbf{z}_i^t$ represents the i-th target domain feature vector, $\mathbf{V}^s$ is the matrix of top-K eigenvectors from the source domain, and $\boldsymbol{\mu}^s$ is the mean of source domain features.

The SSA loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{TTA}}(\phi) = \sum_{d=1}^{K}\{&D_{\text{KL}}(\mathcal{N}(0,\lambda_d^s)\|\mathcal{N}(\tilde{\mu}_d^t,\tilde{\sigma}_d^{t2}))+\\ &D_{\text{KL}}(\mathcal{N}(\tilde{\mu}_d^t,\tilde{\sigma}_d^{t2})\|\mathcal{N}(0,\lambda_d^s))\}\end{aligned} \tag{6}$$

where K represents the number of top eigenvectors used for projection and λ_d^s are eigenvalues from the source domain. The terms $\tilde{\mu}_d^t$ and $\tilde{\sigma}_d^t$ represent the mean and standard deviation of projected target features, while D_{KL} represents the Kullback-Leibler divergence between two Gaussian distributions. Note that we ignored the importance weights α_d from the original paper in our experiments, as they were derived for linear regression layers, while our model uses heatmap regression with convolution layers and softmax.

Due to memory constraints with large feature dimensions ([2048, 8, 8] and [2048, 8, 6]) in body_backbone and hand_roi_net respectively, we had to adapt our approach to feature alignment.

We explored two approaches: using spatial pooling (1x1) to reduce feature size to [2048, 1, 1], and computing separate covariance matrices for each spatial position [2048, i, j]. Note that, spatial position covariance matricies were only calculated for body features because of the limited computational resources. Computation of source statistics were done on all of the images in COCO-WholeBody, and InterHand2.6M datasets.

3.3 Evaluation Framework

To comprehensively assess model performance, we developed a flexible evaluation framework that incorporates multiple complementary metrics. The framework is built around a custom PyTorch Dataset [3] implementation that manages both training and evaluation data, featuring a specialized HandLandmarks class for efficient storage and manipulation of hand keypoint annotations and images. This dataset class can perform self-evaluation with any trained model, generating comprehensive performance reports with configurable metrics. The framework supports custom annotation formats and provides built-in visualization capabilities for qualitative analysis, allowing researchers to visually inspect prediction quality alongside quantitative metrics. Evaluation framework was also used during training of the models, to evaluate validation dataset using PCK and IOU metrics after each epoch, in most of the graphs provided later adaptive PCK was chosen as the primary metric. The complete implementation of these metrics and evaluation framework is available in our public GitHub repository [3].

PCK Metrics. Our primary evaluation metric is the Percentage of Correct Keypoints (PCK), which we implement in two variants. Following [25], the first variant considers a prediction correct if its distance from the ground truth is within $\alpha = 0.05$ of the image size. Additionally, we introduce an adaptive PCK threshold that scales with hand size and pose. While previous work [13] normalizes based on the distance between specific finger landmarks (e.g., between middle and lower points of the middle finger), our approach uses the minimum distance between any two ground truth keypoints as the normalization factor. This adaptive threshold ensures fair comparison across varying hand scales and poses, particularly important for our domain adaptation scenario.

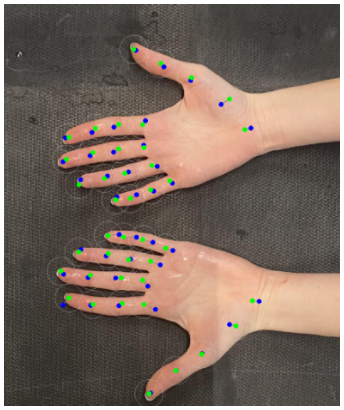


Figure 4: Example visualization produced by our evaluation framework showing hand landmark predictions (shown in blue) connected by lines to corresponding ground truth annotations (shown in green).

The framework includes a visualization module that overlays predicted landmarks with ground truth annotations, color-coding points based on their PCK accuracy (as shown in Figure 4). This visual feedback helps in identifying systematic errors and understanding model behavior across different hand poses and lighting conditions.

Bounding Box Evaluation. To evaluate the accuracy of hand detection, we compute the Intersection over Union (IoU) between predicted and ground truth bounding boxes for both hands. The final score is the average IoU across both hands per image. For cases where ground truth annotations lack explicit bounding box information, we derive boxes from keypoint coordinates by computing the minimum rectangular region containing all landmarks. The IoU is calculated as:

$$IoU = \frac{|B_{pred} \cap B_{gt}|}{|B_{pred} \cup B_{gt}|} \tag{7}$$

where B_{pred} and B_{gt} represent the predicted and ground truth bounding boxes respectively.

4 Experimental Setup and Results

4.1 Implementation Details

Hardware and Software Environment. Our experiments were conducted on a single GPU-enabled workstation equipped with an RTX4060 (8GB VRAM).

Table 1: Performance comparison across different adaptation methods

Method	Full Dataset			Cleaned Dataset		
	IOU	PCK@0.05	APCK	IOU	PCK@0.05	APCK
InterWild (baseline)	0.625	0.656	0.498	0.718	0.777	0.650
AdaBN	0.654	0.676	0.524	NA	NA	NA
Deep CORAL (hand)	0.675	0.741	0.585	0.763	0.840	0.705
SSA (body)	0.646	0.693	0.543	0.710	0.753	0.635
Deep CORAL + AdaBN (best)	0.682	0.756	0.596	NA	NA	NA
SSA (body) + AdaBN (best)	0.656	0.704	0.553	NA	NA	NA
SSA (body) + DeepCORAL (hand)	0.717	0.780	0.617	0.787	0.874	0.727

The implementation utilized PyTorch (2.5.1+cu124) [16] and Python 3.10.11 for model training and evaluation.

During development, we extensively tuned hyperparameters for both AdaBN and Deep CORAL approaches.

For AdaBN, this included optimizing batch size, BN momentum, dataset splits, and BN layer train mode activation criteria.

The Deep CORAL implementation required careful tuning of CORAL loss weight, spatial pooling kernel sizes, learning rate, batch size, data splits, number of epochs, and layer freezing strategies.

Datasets. Our experiments utilized three main datasets:

- **Source Domain Dataset:** For domain adaptation, we used two large RGB datasets: COCO-WholeBody [9] (∼20GB) and InterHand2.6M [15] (∼160GB). These datasets provided source domain statistics for methods such as Deep CORAL and SSA.
- **Target Domain Dataset:** We utilized 5000 IR frames extracted from medical videos [10], sampled at 1 FPS to ensure diversity. Approximately 500 frames were removed during manual cleaning to exclude transitions between patient hand switches, resulting in 4500 frames containing clear hand presentations. For training scenarios involving the hand module, we generated bounding box annotations using Grounding DINO [12]. The consistent camera positioning and standardized hand placement allowed us to optimize the detection process by analyzing half-frames, which also provided reliable left/right hand classification.
- **Labeled RGB+IR Dataset:** We created a validation dataset of 160 images (80 IR and 80 corresponding RGB) using 5 subject hands. The collection included systematic variations in:
  - **Temperature Conditions:** Eight cooling patterns: (1) no cooling, (2) hands briefly submerged in water, (3-7) 1 to 4 fingers cooled for 30s, and (8) whole hand cooled for 30s
  - **Backgrounds:** Two different surfaces: rubber mat and plastic tabletop

  Additionally, we created a separate cleaned dataset of 25 images where all fingers were clearly visible and no cooling was applied. This subset allowed us to evaluate model performance under optimal conditions without the challenges introduced by temperature variations.

All annotations follow a standardized JSON format (see Appendix A for details).

4.2 Results and Observations

Baseline Performance (No Adaptation). Our baseline model (*InterWild* without domain adaptation) was initially evaluated on a set of 80 IR images. For a constant threshold PCK@0.05, we recorded a baseline PCK of 0.608, and an IOU of 0.625, but on a cleaned dataset it reached 0.718 and 0.777 respectively, which shows that interwild perfroms better even on infrared data if the hands are visible.

AdaBN. AdaBN provided moderate improvements of about +3%. For IOU, baseline performance was sometimes slightly improved, although not significantly. For a constant threshold PCK, the best AdaBN configuration reached 0.644 with an IOU of 0.654. Hyperparameter tuning have shown that adding additional target domain data (which was 5000 frames from videos) in addition with data which the model will be tested only decreased the results, as well as manual update of the BN statistics after inference on entire dataset, outlining the most important hyperparameters which are BN layer momentum parameter and batch size.

Deep CORAL. We conducted domain-adaptive experiments with Deep CORAL, focusing primarily on the hand keypoint detection module while using ground truth bounding boxes during training. This decision was motivated by our observation that hand localization in the IR domain is relatively straightforward, with hands being the dominant objects in each frame. This was validated by the high accuracy of the Grounding DINO model (trained on RGB) to produce bbox annotations for IR images. However several experiment were conducted on training body module as well, achieving an IOU of 0.57 (compared to 0.55 baseline) acting as a proof that improvement is possible.

For the keypoint detection training, we implemented Deep CORAL under several hardware-imposed constraints. We used a 1×1 kernel for average adaptive pooling [1] and restricted the batch size to 12 samples. Following the original paper's recommendations, we unfroze all layers in the hand_roi_net module while keeping other layers frozen, and set the learning rate of the last layer (layer4) 10 times higher than other layers (0.001 vs 0.0001). We also experimented with freezing additional layers, though this did not yield better results. The training process jointly optimized the original InterWild keypoint loss and CORAL loss.

Figure 5 shows the training progression through three key metrics. The `joint_img_loss` represents the original InterWild loss for keypoint detection accuracy on the source domain, while `hand_coral_loss` tracks the CORAL loss during training. Additionally, we monitored `validation_hand_coral_loss`, which calculates CORAL loss on unseen validation data. An interesting pattern emerges in these curves: the CORAL loss rapidly decreases in the initial epochs, regardless of the chosen coral_weight and learning rate parameters. This behavior likely stems from the significant size and diversity disparity between our source and target domains. The model quickly learns to align the limited target domain features with the source domain, after which the joint_loss becomes the dominant training signal. This is reflected in the validation curves (Figure 6), where accuracy initially decreases as the model adapts to the domain shift, then gradually recovers as it continues training with the already-aligned feature representations.

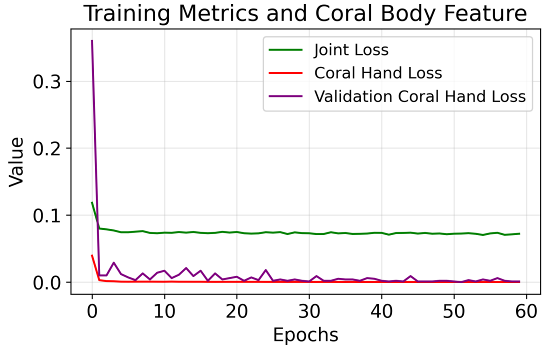


Figure 5: Training Metrics and Coral Body Feature alignment over 60 epochs. The joint_img_loss (green) represents the original InterWild loss for keypoint detection, hand_coral_loss (red) tracks the CORAL loss during training, and validation_hand_coral_loss (purple) shows CORAL loss on unseen validation data. The consistent reduction in validation CORAL loss indicates successful feature alignment between domains.

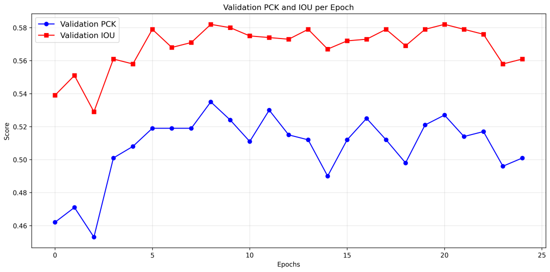


Figure 6: Validation PCK (IR subset of 80 images) and IOU for Deep CORAL over 60 epochs.

SSA Method. We then investigated the *Subspace Alignment* (SSA) approach, which trains primarily on the target domain. By doing so, and as the original SSA paper [7] suggests, we were able to freeze most of the model parameters except for the BN layers, thus allowing batch sizes as large as 64. Larger batch sizes not only accelerate training but also provide better estimation of target domain statistics, since SSA loss is based on the mean and variance of the batch.

Analysis of feature spaces revealed interesting sparsity patterns in our model's representations (Table 2). The source domain statistics were computed through a two-pass process over the combined MSCOCO and InterHand2.6M datasets (approximately 200GB). The first pass calculated mean feature values across both body and hand features using the full feature space. The second pass computed the global covariance matrix using previous computed mean values iteratively, making use of spatially pooled features of size [2048, 1, 1]. Analysis of resulting covariance matrices revealed that the body_backbone maintained full rank with all 2048 dimensions having significant variance, the hand_roi_net exhibited substantial sparsity with only 1265 valid dimensions (variance $> 1e-10$). This observation aligns with SSA's core premise that regression features often concentrate in lower-dimensional subspaces, particularly evident in the hand_roi_net's compact 7-dimensional subspace compared to body_backbone's 175 dimensions.

Table 2: Feature space dimensionality analysis for SSA

Module	Total dims	Valid dims	Subspace dims
body_backbone	2048	2048	175
hand_roi_net	2048	1265	7

We started by adapting the **body_backbone** portion of the model (accounting for around 0.16% of total parameters). An AdamW optimizer was employed with a weight decay of 0.01. Two principal SSA hyperparameters were explored:

1. The subspace dimension K retained during the alignment, where literature suggests $K = 100$ as a highly effective choice.
2. The learning rate schedule.

In our experiments, $K = 100$ was not always optimal, so we tested multiple values. Figure 7 (training) and Figure 8 (validation) demonstrate a clear positive trend, with validation performance typically converging after 1–3 epochs. Through SSA on just the body_backbone, we achieved an adaptive PCK of 0.543 (+4% from the baseline of 0.50) and an IOU of 0.646 (+0.02 from baseline of 0.625).

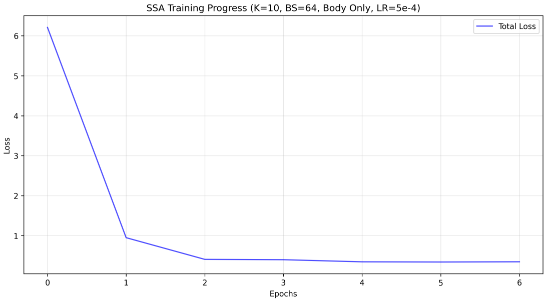


Figure 7: SSA training curves (body_backbone). Loss converges within the first few epochs.

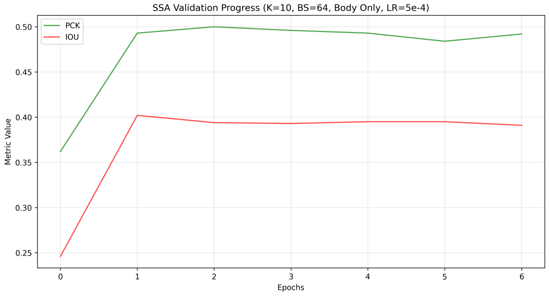
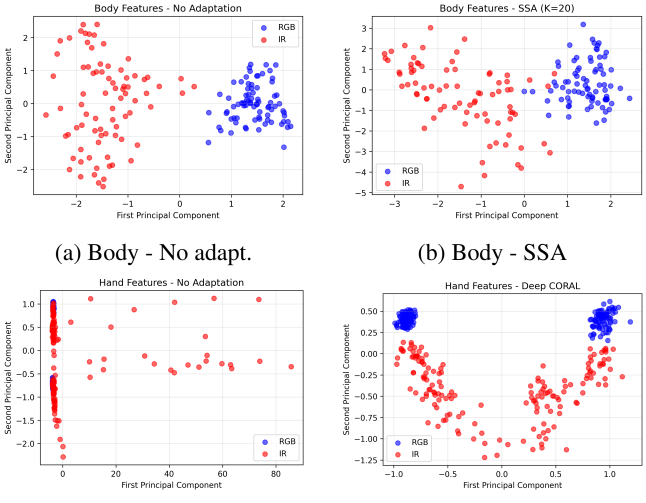


Figure 8: SSA validation curves (body_backbone). Validation accuracy typically stabilizes after 1–3 epochs, yielding about 0.50 PCK and 0.40 IOU.

To further investigate spatial constraints, we considered computing separate covariance matrices for each spatial location of the feature map (e.g., a $2048 \times 8 \times 6$ feature map leads to 48 covariance matrices). The aggregated SSA losses did not improve performance, potentially due to the large number of subspace alignment objectives. Training curves with this spatial extension can be seen in Appendix B.

Lastly, we explored adapting the **hand_roi_net** module using a similar SSA pipeline, starting from the best checkpoint for the body_backbone. Various subspace dimensions (K values from 5 to 300) and learning rates (ranging from 1×10^{-5} to 5×10^{-2}) were tested, but no improvement over baseline was observed (see Appendix C for training curves). A likely reason is that *hand_roi_net* receives bounding boxes (RoIs) that are already suboptimal in IR data, so optimizing BN layers alone may be insufficient. The experiments feeding hand_roi_net with ground truth bboxes from RGB Data also have not shown any improvements, although proper hyperparameter tuning could potentially improve the results.



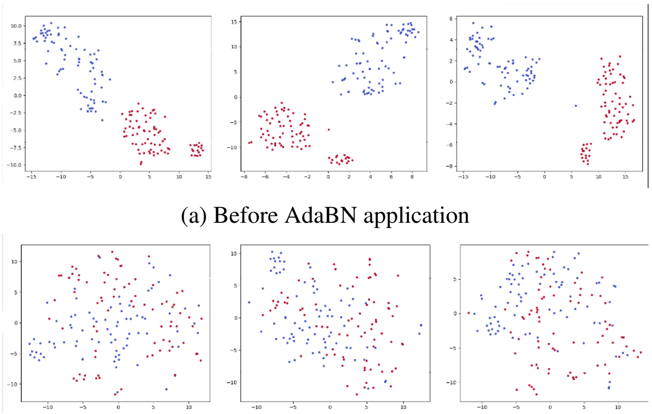
(a) Body - No adapt.

(b) Body - SSA

(c) Hand - No adapt.

(d) Hand - CORAL

Figure 9: PCA vis. Red: IR, blue: RGB.



(a) Before AdaBN application

(b) After AdaBN application

Figure 10: t-SNE mapped 2-dimensional representation of feature distributions from the last three batch normalization layers in hand_roi_net module. The x and y axes show t-SNE's first and second components (t-SNE 1 and t-SNE 2) respectively. Red points represent target domain (IR) features, while blue points represent source domain (RGB) features. The visualization demonstrates how AdaBN helps align the feature distributions between domains.

Feature Space Visualization Analysis. To further validate the effectiveness of our adaptation methods, we conducted feature space visualizations comparing both domains before and after applying each adaptation technique.

Figure 9 presents PCA visualizations using the two most significant components of feature outputs from both the hand_roi_net (for Deep CORAL) and body_backbone (after SSA) modules. The visualizations demonstrate clear evidence of feature alignment, though the nature of this alignment differs between methods. While both domains' features show increased proximity post-adaptation, they maintain distinct clusters rather than complete overlap.

This partial separation can be attributed to our use of average adaptive pooling, which, while necessary due to computational constraints, resulted in some information loss that prevented complete feature space alignment.

In contrast, Figure 10 shows t-SNE visualizations of feature distributions in the final batch normalization layer of hand_roi_net before and after applying AdaBN. These visualizations reveal nearly complete feature alignment between source and target domains. However, despite achieving the strongest feature correlation among our methods, AdaBN's performance improvements were more modest compared to other approaches. This observation suggests that while adapting batch normalization statistics is beneficial, comprehensive domain adaptation requires modifications beyond just the BN layers.

Ensemble of Methods and Performance Analysis. We evaluated our adaptation methods both individually and in combination, testing on both the full IR test set and a cleaned subset (where all fingers are clearly visible). The comprehensive results are presented in Table 1.

AdaBN alone showed modest improvements (+2-4%) but proved unstable, requiring extensive hyperparameter tuning across batch sizes and momentum values. Its effectiveness diminished when combined with other methods, likely because both CORAL and SSA inherently update batch normalization statistics during training. Additionally, AdaBN's evaluation on our limited IR dataset may have constrained its potential.

The most effective approach combined SSA and Deep CORAL methods, which target complementary parts of the model (body_backbone and hand_roi_net respectively). This combination yielded an 11% average improvement over baseline, surpassing all other configurations. Adding AdaBN to this ensemble provided no additional benefits, as both major model components were already well-adapted to the target domain via BN statistics refinement during training.

For evaluation, we used two PCK variants: PCK@0.05 (fixed threshold at 5% of image size) and APCK (adaptive threshold based on minimum inter-keypoint distance). The baseline InterWild model showed significantly better performance (+9%) on the cleaned dataset compared to the full dataset, highlighting the impact of finger visibility on model performance. However, the relative improvements from our adaptation techniques remained consistent across both datasets, suggesting that our adapted models successfully address both the RGB-to-IR domain shift and demonstrate improved robustness to finger occlusions and visibility challenges.

5 Discussion

Our results show that lightweight domain adaptation methods can meaningfully improve IR hand landmark detection without exhaustive retraining. In particular, we observed that aligning earlier feature extraction layers (the `body_backbone`) with Subspace Alignment (SSA) addresses much of the IR domain shift, while adapting hand-specific modules remains limited by imperfect bounding boxes. Notably, combining SSA on the `body_backbone` and Deep CORAL on the `hand_roi_net` brought an 11% gain in PCK over the baseline. This pattern highlights the importance of module-specific strategies for domain adaptation in tasks where errors propagate from body to hand levels.

Adopting simpler approaches also paid off: AdaBN alone improved PCK by roughly 3–4% with minimal GPU memory usage. However, its impact diminished when paired with more complex methods that inherently update batch-normalization statistics. Despite hardware constraints restricting our batch sizes—undermining some theoretical advantages of Deep CORAL—both Deep CORAL and SSA demonstrated effectiveness when carefully tuned. Compared to the original model, these methods placed less emphasis on large-scale source training and more on fitting IR-specific distributions through adjustment of key parameters.

In analyzing cleaned IR images (where fingers are fully visible), we found that a significant fraction of the performance gap stems from physical limitations: "thermal vanishing" occurs when finger temperature aligns with ambient conditions. This effect underscores that even the best-adapted models cannot fully recover missing thermal cues. As a result, our findings confirm that external factors such as visibility of hands— beyond modeling choices—impact IR performance.

Given the results here, we recommend modular adaptation pipelines, where early layers undergo thorough domain alignment and later layers incorporate specialized corrections or complementary sensing to mitigate thermal vanishing issues.

Limitations The most critical limitation of this study was hardware constraints, specifically our 8 GB VRAM capacity, which prevented full model training for more advanced methods like Deep CORAL. As a result, we only trained separate modules and could not investigate or align the entire feature space due to the prohibitive size of the covariance matrix. Additionally, these memory constraints impeded thorough hyperparameter tuning for Deep CORAL and disallowed larger batch sizes, prolonging the experimental process even though our dataset was relatively small (5,000 images). Another major limitation lies in the availability and diversity of the infrared data itself. The frames—extracted at a rate of 1 fps from videos where hands always remained in the same position—did not provide the necessary variability to adapt the model to a wider range of infrared hand detection scenarios. Furthermore, the evaluation of our adaptation methods (presented in Table 2) was conducted on only 80 infrared images annotated by our research team, which is not a sufficiently large sample size to robustly validate the reported improvements in model performance.

6 Conclusion and Future Work

Most existing domain adaptation techniques like AdaBN [11] and Deep CORAL [18] were developed several years ago and Deep CORAL [18] paper experiments primarily focused on classification tasks (thus tailored differently in their implementation details), whereas our work requires a regression-oriented approach for hand landmark detection with heatmap regression.

Although recent methods such as SSA [7] from 2024 specifically address domain adaptation in regression settings, they do not incorporate the latest advanced deep learning innovations, particularly attention-based techniques that have sparked interest in medical and computer vision applications [19].

Moreover, large-scale synthetic IR data generation and augmentation strategies—inspired by works that leverage synthetic data for more robust training [14] and guided by IR scene simulation methodologies [21]—could significantly enhance model performance. Specifically, combining synthetic IR images with real data, collected under diverse conditions, would yield a broader distribution of thermal hand poses for training.

In addition, prior work such as RegDA [8] has shown promise for unsupervised domain adaptation in keypoint detection (also leveraging heatmap regression, which is used in InterWild). Integrating these insights, alongside deeper networks with attention, likely offers a fruitful direction for achieving robust, clinically viable IR-based hand landmark detection.

Overall, while shallow domain adaptation shows promise, deep methods require substantial computational and data resources. Future work will focus on bridging this gap to deliver clinically viable IR-based hand landmark detection.

7 Responsible Research

The study implements ethical considerations through informed consent processes where participants were explicitly asked about their willingness to include hand images in a publicly available dataset. Data collection followed strict anonymization protocols, with all images undergoing masking procedures to protect identities. While the research motivation originated from leprosy detection needs, the technical focus specifically addresses domain shift challenges between RGB and infrared modalities for hand keypoint prediction under constrained computational resources and small target domain datasets. Any medical applicability claims would require additional validation through larger clinical studies involving certified practitioners and more comprehensive evaluation frameworks. For transparency, the complete experimental codebase - including training implementations, evaluation framework, and data visualization tools - will be archived in TU Delft's institutional repository and is publicly available on GitHub [3], utilizing only open-source datasets and model architectures. The final model weights demonstrating optimal performance through combined adaptation methods (AdaBN and Deep CORAL) will be published to enable result verification, though potential result variations may occur due to unset random seeds during training. LLM tools (Claude Sonnet 3.5) were employed solely for text summarization and grammatical corrections, with no algorithmic or conceptual contributions from generative AI systems.

A Dataset Annotation Format

The keypoint annotations in our dataset follow a standardized JSON format as shown below:

```
[{
    "image": "image_name.jpg",
    "width": 1080,
    "height": 1440,
    "landmarks": [
        [ // First hand
            {"x": 0.5744, "y": 0.6187}, // keypoint
            // ... 21 keypoints total
        ],
        [ // Second hand
            {"x": 0.4343, "y": 0.5856}, // keypoint
            // ... 21 keypoints total
        ]
    ],
    "normalized": true
}]
```

B Additional SSA Training Curves

C SSA Training Curves for Hand ROI Net

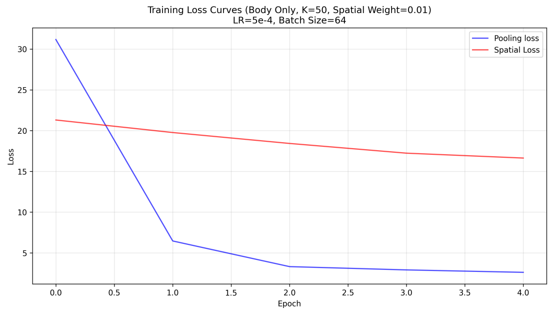


Figure 11: SSA training curves with spatial covariance matrices. Computing 48 additional losses proved computationally heavy.

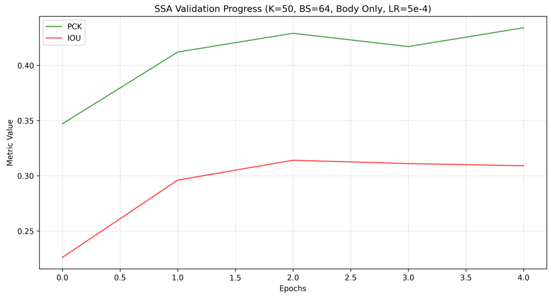


Figure 12: SSA validation curves with spatial covariance approach. No improvement was observed compared to the simpler SSA variant.

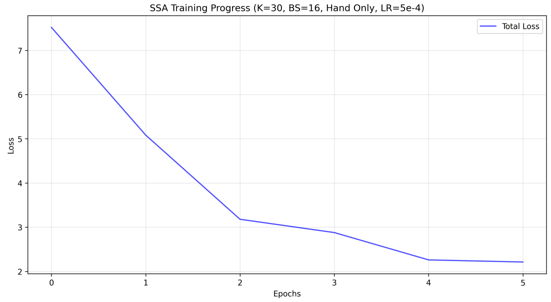


Figure 13: SSA training curves for the hand_roi_net module. Significant parameter unfreezing required, reducing batch size to 16 due to VRAM constraints.

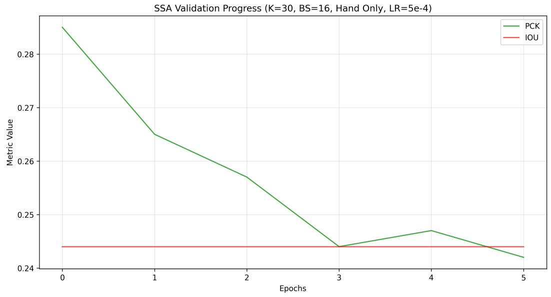


Figure 14: SSA validation curves for the hand_roi_net module. No improvement was observed, possibly due to suboptimal bounding box inputs.

References

[1] Pytorch documentation: AdaptiveAvgPool2d. https://pytorch.org/docs/stable/generated/torch.nn.AdaptiveAvgPool2d.html. Accessed: 2024.

[2] Pytorch documentation: BatchNorm2d. https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html. Accessed: 2024.

[3] EraChanZ. Research project: Domain adaptation for hand pose estimation. https://github.com/EraChanZ/RP, 2024. GitHub repository.

[4] Facebook Research. Interwild: 3D interacting hands recovery in the wild. https://github.com/facebookresearch/InterWild, 2023. GitHub repository.

[5] Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: A survey. *IEEE Transactions on Biomedical Engineering*, 2021.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[7] Yihua Jiang, Jiawei Ren, Jianfei Gu, and Yong Jiang. Test-time adaptation for regression by subspace alignment. *arXiv preprint arXiv:2410.03263*, 2023.

[8] Yihua Jiang, Jiawei Ren, Haoxuan Sun, Jianfei Gu, and Yong Jiang. Regressive domain adaptation for unsupervised keypoint detection. *arXiv preprint arXiv:2103.06175*, 2021.

[9] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[10] A. Knulst. Personal communication, 2024. Medical video dataset for hand pose estimation research.

[11] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[13] MathWorks. Hand pose estimation using HRNet deep learning. https://www.mathworks.com/help/vision/ug/hand-pose-estimation-using-hrnet-deep-learning.html, 2024. Online documentation.

[14] Gyeongsik Moon. Bringing inputs to shared domains for 3D interacting hands recovery in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[15] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *European Conference on Computer Vision (ECCV)*, 2020.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[17] Irene Schemkes. Semi-automatic temperature analysis based on real-time hand landmark tracking in infrared videos. Master's thesis, Delft University of Technology, 2023.

[18] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, 2016.

[19] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Attention mechanisms in computer vision: A survey. In *Computer Vision – ECCV 2020*, pages 442–458. Springer, 2020.

[20] Wikipedia contributors. Domain adaptation. https://en.wikipedia.org/wiki/Domain_adaptation, 2024. Accessed: 2024.

[21] Bin Zhang, Yuncai Wang, and Wei Liu. Infrared imaging model for scene simulation and its validation. *Infrared Physics & Technology*, 67:531–536, 2014.

[22] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. In *arXiv preprint arXiv:2006.10214*, 2020.

[23] Xin Zhao. Awesome domain adaptation. https://github.com/zhaoxin94/awesome-domain-adaptation, 2024. GitHub repository.

[24] Jinghao Zhuang. Mediapipe pytorch. https://github.com/zhuang-jia-xu/mediapipe-pytorch, 2020. GitHub repository.

[25] Christian Zimmermann and Thomas Brox. Percentage of correct keypoints (pck) metric for hand pose estimation. *arXiv preprint arXiv:2103.06175*, 2021.