SHALINI MAITI, B.Tech (ICT)

Domain Randomization for Hand Pose Estimation

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Technische Universität, Graz

Supervisor

Professor Vincent Lepetit

Mentor

Dr. Mahdi Rad

Institute for Computer Graphics and Vision

Graz, December 2020

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Acknowledgement

I am grateful to my parents, Sudipta and Kalpesh and my sister, Sayantani for being my backbone throughout the journey that's culminated in this work. They have always been my own, personal support system.

I would like to thank my mentor, Dr. Mahdi Rad and my supervisor Professor Vincent Lepetit, for taking time out of their busy schedule to guide me and provide me with important critique and constructive feedback for the entire duration of the thesis.

I would also like to express my thanks to the Lepetit team at the ICG, especially Shreyas Hampali, for providing me with technical help and resources wherever required. Special thanks to Dr. Peter Roth for arranging a server to run computations and also to Elisabeth Tomacelli for making my move to Graz smooth and making my stay here fun!

I would like to thank all my mentors during my master's programme - Professor Denis Kalkofen, Professor Friedrich Fraundorfer, Professor Thomas Pock and Professor Schmalstieg - who gave me opportunities, and work tirelessly to make life easier for us, the students. I am thankful also to Dr. Shohei Mori, Dr. Patrick Knöbelreiter and Dr. Shaifali Parashar; who were instrumental to any contributions that I have made, shaped my work ethic and patiently answered many, many questions.

Lastly, I'd like to thank both sets of grandparents, the Maitis and the Nandis, for teaching me that learning can come from everywhere and at all times, if one is humble and curious.

Abstract

Hand Pose Estimation from colour images is one of the challenging tasks in Computer Vision. Recently many Deep Network-based approaches have been proposed to tackle this problem. However, they mostly rely on real RGB images. Acquiring real RGB images can be very time consuming, cumbersome and error-prone. In this work, we therefore propose to train Deep Network-based models for Hand Pose Estimation using only synthetic images. Using synthetically generated images is extremely beneficial as it is easy to create a virtually infinite training set made of such images along with their annotations, which covers nearly all feasible poses, along with variations in texture, backgrounds, random lighting conditions, etc. However, synthetic images do not exactly look like real images and result in suboptimal performance due to Domain Gap. To bridge this gap, we use the Domain Randomization technique and provide an extensive ablation study to evaluate the influence of different steps in our pipeline. Finally, we demonstrate our approach on the publicly available datasets such as Freihands, STB and Large-scale Multiview 3D Hand Pose Dataset.

Abbreviations

1. DoF : Degress of Freedom
2. RGB : Red, Green and Blue channels in an image
3. RGBD : Red, Green, Blue and Depth (of the objects) in an image
4. CNN or ConvNet: Convolutional Neural Network
5. PCA : Principal Component Analysis
6. SMPL model : Skinned Multi-Person Linear model
7. MANO model : hand Model with Articulated and Non-rigid defOrmations
8. CPM : Convolutional Pose Machines
9. LBS : Linear Blend Skinning
10. t-SNE : t-distributed Stochastic Neighbor Embedding
11. 2DKPE : 2-Dimensional key point error
12. AU2D : Area under the 2D error curve
13. BD : Base Dataset without any hand shape and lighting variations
14. BD-S : Base Dataset with hand shape variation but without any lighting variation
15. BD-SL : Base Dataset with both hand shape and lighting variations
16. BG : Background

Contents

List of Figures

1 Introduction

Hand Pose Estimation is a computer vision task where the pose of the hand, i.e, the parameters that define the hand is predicted from images. It could include rotation of the angles, shape, size and other characteristics of the hand. Our work focuses on predicting the 2D positions of the joints of the hand. Hand Pose Estimation has applications in Augmented/Virtual/Mixed reality and Human Computer Interaction because humans use their hands for most of the essential tasks - for tactile interaction or gestures. Many CNNs predict the Hand Pose with high accuracy; such as [34], [67].

Training Deep Networks are highly data-driven and require data in the order of tens of thousands to learn the parameters of a model (the Hand Pose, in our case). Gathering and annotating these training datasets, which is usually done manually, is error-prone and demands a lot of time and precision. A solution to this problem is to use synthetic images for training. This removes human involvement from most of the data generation process, saves time and reduces training error caused due to imprecise labelling. This was the main motivation behind this work.

However, a model trained on synthetic data performs poorly on real data because of the *Domain Gap* between the two datasets. Simulating exact physical properties of the real world in synthetic image generation pipelines is very complex and difficult to achieve. We attempt to reduce this gap by using a technique called *Domain Randomization*. This involves introducing randomness or variance into the training data through the parameters of the hand and the image. Our goal is to use only synthetic data for Hand Pose Estimation from RGB images and to bridge the Domain Gap using Domain Randomization.

We use the MANO hand model renderer [45] to render a variety of datasets consisting of synthetic RGB images of the hand, the OpenPose architecture [7] to train models on these synthetic datasets and evaluate our approach on publicly available real datasets. Additionally, we used t-SNE to ascertain a sufficiently varied pose distribution of our datasets with respect to the test datasets used in our work. We also conduct ablation studies and make some observations about the correlation between the distribution of different dataset parameters and the performance of the trained models.

2 Related Work

We bridge the Domain Gap using Domain Randomization to make training of a Deep Network using purely synthetic images possible. So, in this section, we provide a body of work consisting of traditional and state-of-the-art Hand Pose Estimation solutions as well as Deep Learning solutions with similar motivation as ours, i.e, to alleviate the problem of procuring and annotating real datasets.

2.1 Hand Pose Estimation

There is a long body of work that attempts to solve the problem of Hand Pose Estimation. We refer to three literature reviews of this task [13, 12, 29], that provide a comprehensive history and development of solutions. Some early solutions include magnetic sensors such as *gloves* [51] or *visual markers* [36] to detect fingertips or joints. These are very effective but are intrusive and restrict natural interactibility. Other approaches include 3D tracking methods using feature-matching of geometric primitives and some reliance on prior knowledge. These methods are unreliable in practice, especially on cluttered backgrounds or rapidly moving hands because there is a high likelihood of making inaccurate matches. Since the difference between the input space of image pixels (order of millions) and the output space of pose parameters (order of tens) is so vast; algorithmic, intuitive or other prior-based methods do not perform as well as mapping using data-driven approaches such as in Machine Learning, such as, *Random Decision Forests* [28, 48], which were popular methods in the pre-Deep Learning era. Some contemporary methods using Deep Learning to estimate Hand Pose from depth maps are [34, 9, 61, 57]. Depth cameras are expensive and less ubiquitous, making these systems less favourable. Similarly, some state-of-the-art approaches using Deep Learning to estimate Hand Pose from a single RGB image include [67, 21] and methods to predict Hand Pose and shape include [15, 3]. These approaches require a huge amount of real annotated data and capturing this data is time-consuming and error-prone.

2.2 Training with synthetic data, Semi or Unsupervised Learning

GAN-erated Hands [39] extends *Cycle-GAN* [66] to propose a 3D hand tracking solution with unpaired real and synthetic images as input using two network pairs, which are jointly trained to generate realistic synthetic images from the real input. The first network pair takes a real image and transfers that to a synthetic image, i.e, real2synth and the second one takes this synthetic image and transfers this to the real domain, i.e, synth2real. In [17], a sufficiently diverse and realistic dataset *ObMan Dataset* is generated which consists of hand and object interactions through 3D reconstruction of hands and object from real RGB data for the purpose of Hand and Object Pose Estimation. *HOPS-Net* [25] uses *Augmented Cycle GAN* to make synthetic training images seem more photorealistic for the purpose of pose and shape estimation of handheld objects. In [58], a Self-Supervised method pre-trained with a synthetic hands datasets is used to predict the 3D pose of the hand. Similarly, [11] also uses synthetic data for pretraining followed by fine tuning on unlabelled data using a loss combining depth, collision and physical components. This removes the requirement of manually annotating data. In [43], a domain adaptation method is used to minimize the Domain Gap error. In [49], Simulated+Unsupervised Learning is used. The simulator produces realistic using GANs with synthetic input images which are subsequently refined to improve its realism using a *refiner network* using unlabelled real data. The above mentioned works have proposed methods to produce photorealistic images to bridge the Domain Gap. In contrast to the above works, we use only synthetic images for training. We do not attempt to generate photorealistic images and we apply our method to the task of Hand Pose Estimation.

2.3 Domain Randomization

Several researchers have attempted to use Domain Randomization in order to bridge the domain gap. In terms of physical modelling, in [38], a policy is trained on an ensemble of dynamic simulated models in order to transfer to real world robot exploration. In [2], physical characteristics like friction are randomized in order to make the robot more robust to modelling errors. In [63], a model of varied physics is trained and [44] attempted to train a policy by exploring and combining various training algorithms. Both approaches did not transfer well to the real world. In computer vision, some uses of Domain Randomization can be seen in works like where models trained on synthetic images were augmented using 3D models for object detection [52, 41]. Many domain transfer approaches try to model synthetic data to appear very similar to the real world scenario. However, [54] works on the interesting hypothesis that by introducing enough variability instead of simulating realistic images, the network exhibits robustness in real-world scenarios. They achieved success in the task of object localization, a very important task in the realm of robotic manipulation.

2.4 Our Contribution

We introduce a similar type of Domain Randomization as [54] to the task of Hand Pose Estimation. We focus on the introduction of randomness through various model and image parameters, instead of synthesizing photorealistic images. We use MANO model renderer [45] to render RGB images of the hand. We randomly change various parameters such as lighting, background, hand texture, noise, hand parameters etc, train them using OpenPose [7] and analyze the results of randomising different parameters on real world datasets. There is a big improvement in the performance of models trained using purely synthetic datasets with Domain Randomization when compared to the ones trained without it.

3 Theoretical Background

In the following chapter, we present a rigorous overview of all the different segments used in the creation of this work. Section 3.1 explains the anatomy and modelling of the hand. Section 3.2 describes the task of Hand Pose Estimation. Section 3.3 delineates the major challenges faced in Hand Pose Estimation. Section 3.4 elucidates Domain Gap and Domain Randomization. Section 3.5 demonstrates the tools and techniques use to generate synthetic datasets. Section 3.6 details data distribution comparison techniques.
Section 3.7 illustrates Deep Learning architectures for Hand Pose Estimation. Section 3.8 summarizes performance metrics used in Hand Pose Estimation.

3.1 The Model of the Hand

The human hand consists of 27 bones - 19 of them are contained in the fingers and palm, and 8 in the wrist. Together, these bones form a rigid body connected via joints that allow 1 or more degrees of freedom (DoFs). The metacarpophalangeal (MCP) joint connects the fingers to the palm, the Interphalageal (IP) connects the finger segments, more specifically the Distal IP (DIP) connects the topmost segment with the middle segment and the Proximal IP (PIP) connects the middle segment with the bottommost segment. The anatomy of the hand is depicted in Figure 3.1 (a). The kinematic model showing degrees of freedom is in Figure 3.1 (b).

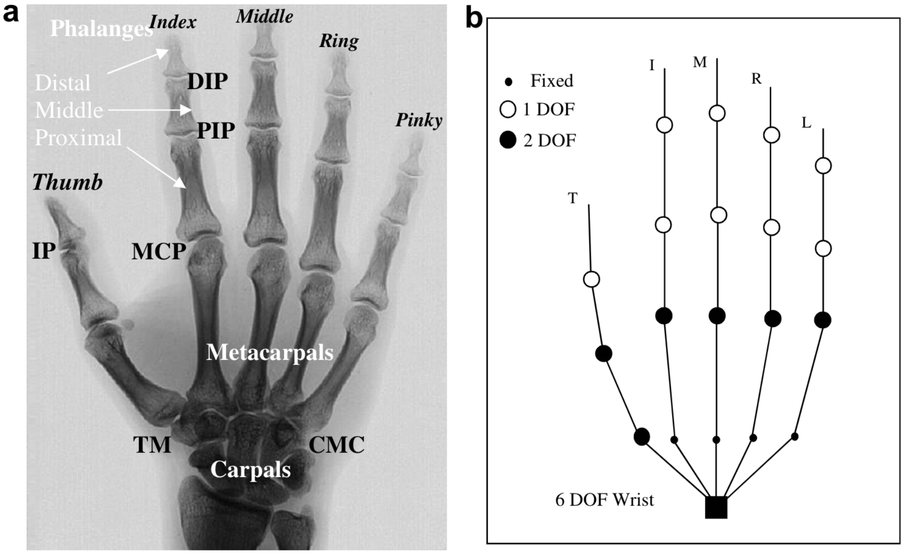


Figure 3.1: (a) Hand anatomy, (b) Kinematic model of the hand. Image credits: [13]

.

Given the fact that the human hand is dexterous, with similarly shaped and coloured fingers and a high number of DoFs; modelling its natural motion and shape is not a trivial task. Several techniques have been used to model the hand including reducing its dimensionality by using inter-joint dependencies or PCA parameters. Postural Hand Synergies for Tool Use [46] observes that two principal components account for over 0.8 of the variance in data. [47] observes that 3-6 principle components covers 0.8 - 0.9 of the pose variance. That covers most of the meaningful pose synergies. Some works have used shape primitives to approximate the shape of the hand [40], or Linear Blend Skinning over a triangulated mesh [30], or used personalized hand models that were tailormade for the user interacting with the system [53]. We later provide a more detailed description of the MANO model [45], which is the hand model that we use to generate hands for our training datasets.

3.2 Hand Pose Estimation

Hand Pose Estimation is an open computer vision problem. Hand Pose defines the structure and articulation of a hand - shape, joint locations, rotation etc. The solution is expected to predict some subset of this output space. In computer vision tasks, this typically includes locations of salient hand joints using either single RGB, depth, stereo or RGBD images. Figure 3.2 shows an example of a successful prediction of 21 joints.

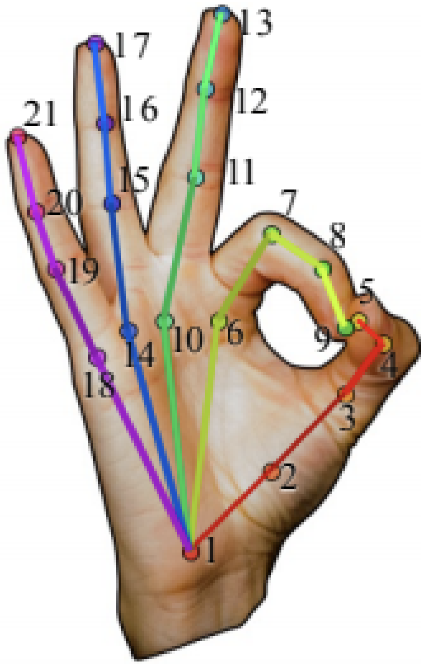


Figure 3.2: Hand Pose Estimation: 2D location of 21 hand joints. Image credits: [50].

Hand Pose Estimation is useful for tracking the hand in real time. This has various applications in Virtual/Mixed/Augmented reality, one of which is shown in Figure 3.3. The advent of Convolutional Neural Networks and increased worldwide funding into domains like Virtual/Augmented/Mixed Reality and Human Computer Interaction has made Hand Pose Estimation a very attractive problem to solve. Some of the areas of application have been listed below:

1. Object Manipulation: Using simple gestures for selection, navigation and manipulation tasks [4].
2. For the design of Command and control Interfaces [[27], [55]].
3. To use in Multimodal UI: Using gestures along with speech and other modalities would be the perfect simulation of innteraction and communication and that is what a full-pose, unconstrained Hand Pose Estimation would help enable.
4. Immersive Virtual Environment applications: Surgical simulations [31], immersive training systems such as the Virtual Glove Box [5].
5. Other systems such as a mid-air keyboard [37], understanding alphabets drawn in the air using fingertip detection [8], sign language interpretor [62] etc.

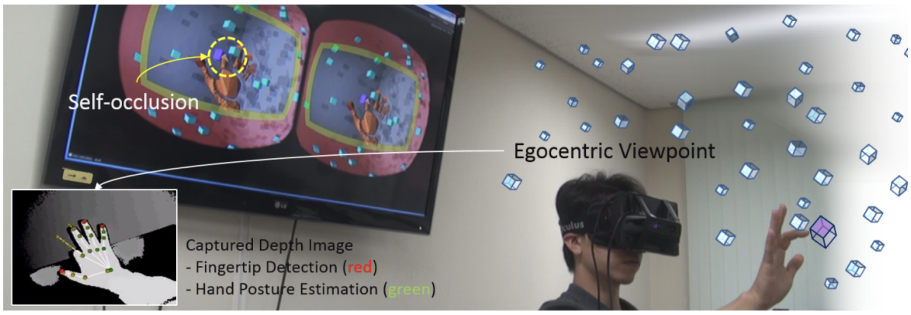

Self-occlusion
Egocentric Viewpoint
Captured Depth Image
- Fingertip Detection (red)
- Hand Posture Estimation (green)

Figure 3.3: 3D Finger Cape 3D [22] is an example of using Hand Pose Estimation in Virtual Reality.

3.3 Major Challenges of Hand Pose Estimation

Some of the major challenges associated with the task of Hand Pose Estimation are as follows:

1. High dimensional problem: Since the hand is an articulated object with 27 DoFS, single-frame prediction of the full (not partial or reduced) Hand Pose for unconstrained dynamic high-level movement, i.e, not just pointing and gesturing, of the hand is a challenge.
2. Difference of orders on the input and output space: The input space is that of the image, which consists of millions of pixels and the output space is in tens of parameters, i.e, the Hand Pose [29].
3. Self-Occlusions: Given that the hand is mostly a concave object, from some camera angles, some of the poses of the hand occlude some of the joints - this is referred to as self-occlusion and can lead to ambiguous inference of the pose.
4. Non-uniform background: Having a non-uniform background filled with non-hand objects, patterns/textures, cluttered with objects etc. add distractions to the network.
5. Different illumination conditions: Different types of lighting, i.e, daylight, coloured light, ambient or different intensity of light.
6. Rapid movement: If the hand is moved faster than the frame rate of the camera, the image would contain motion blur and the pose would be difficult to determine [12].

3.4 Domain Gap and Domain Randomization

When a model is trained entirely on a synthetic training dataset, it does not necessarily perform well on a real test dataset. This is because there are many differences in the characteristics of the two domains - source domain, which is drawn from a synthetic distribution and the target domain, drawn from a real distribution. Synthetic data generation pipelines that attempt to simulate all the properties of the real domain are very complex and demanding on resources. This results in suboptimal performance of such a trained model on a real dataset. This is called Domain Gap, Domain Bias or a Reality Gap. Various solutions have been proposed to solve the Domain Gap problem described earlier [39, 25, 43]. For example, System Identification involves tuning the parameters of the synthetic rendering system to match the real world data. However, this is time consuming and may still have errors, because even photorealistic renderers cannot model many physical effects with the kind of richness as the real world without the help of very complex graphic pipelines. Another problem is the error and noise induced through the sensors in the real world. Domain Randomization is a technique where one introduces variability into the simulated environment by exposing the model to various randomized environments while it is being trained. The idea behind this is that if significant and relevant enough variance is introduced when the model is trained on simulated and is later tested on real data, the real world appears to simply be another variation to the model [54].

3.5 Synthetic Dataset Generation

The model that we use to generate synthetic data is called the MANO model [45]. The acronym stands for hand Model with Articulated and Non-rigid defOrmations. The process of generating this model is two-fold. In the first stage, they collect a large amount of hand scans, specifically around a 1000 scans using a 3dMDhand System [1] in a wide variety of poses following from the grasp taxonomy of [14] with a few additional poses. These scans have a resolution of approximately 50K vertices and accuracy within 0.2mm root mean square error. They were gathered from 31 subjects and iteratively learned a model by using these scans to align a template. Next, they integrate this hand model to a full SMPL body model (M) [33]. This articulated mesh of the model with shape (T_P) and joint locations (J), is a function of the pose ($\vec{\theta}$), hand shape ($\vec{\beta}$) and blend weights (**W**). A Linear Blend Skinning function (W) is applied as a skinning function on this mesh. The formulation of the original SMPL model is as follows:

$$M(\vec{\beta}, \vec{\theta}) = W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathbf{W}) \tag{3.1}$$

$$T_P(\vec{\beta}, \vec{\theta}) = \mathbf{T} + B_S(\vec{\beta}) + B_P(\vec{\theta}) \tag{3.2}$$

Overall, the parameters that are used to define this model include the shape components, blend shape and weights, hand joint locations linearly regressed to a sparse matrix and the template of the hand. To learn these parameters, they choose a linear mapping with PCA parameters to express the kinematic model, shape and global orientation of the model. By definition, this does not cover the full pose space but [46] observes that most hand poses are covered by low-dimensional manifolds, thereby, most of the pose space is covered by varying the values 6 to 12 parameters.

3.6 Dataset Distribution Comparison Metric

There are many data visualization techniques that help us understand distributions and similarities between different data sets. For high dimensional data, it is also important to map them to low-dimensional data outputs that preserve their structure and relationships with other datasets to be plotted in 2D or 3D plots. There are some methods that use linear mapping such as PCA [23]. However, this type of mapping does not preserve non-linear dependencies. To visualise high-dimensional data while preserving their non-linear structural dependencies, using t-SNE [35] as a technique is beneficial. It has a non-linear mapping based on SNE [19] that effectively solves the crowding problem where in higher dimensional surfaces, like a sphere, the surface expands with a much higher rate with the radius as the dimension is bigger so when projected to lower dimensions, these surface points start crowding. The t-SNE technique uses a two-step process to generate a distribution. First, it generates a probability distribution to describe relationships between neighbouring points as a gaussian centered around each point with respect to the other points in the dataset in the high-dimensional space. Next, it uses this distribution to map these relationships into the lower-dimension.

3.7 Deep Learning Architecture for Hand Pose Estimation

3.7.1 Convolutional Neural Networks

Learning complex representations in high dimensional data, such as images, is a non trivial task. Convolutional Neural Networks (CNNs) contain multiple layers that progressively learn representations. They are designed to be able to process multi-dimensional raw input - 2D images or 3D volumetric images [26]. Inspired from the hierarchical organisation of the human visual cortex [20], a CNN is constructed with many interconnected layers of neuron structures. The neuron layers and increasing depth adds complexity to the network. The mathematical formulations of the the convolutional layer neurons are represented by the convolution filters that convolve with the input or output of the images of the previous layer. The results of these convolutions are representative of meaningful, learned features and passed on to the next layer in the feature maps that progress from low-level features starting from the raw input data to high-level abstraction that finally converges to the result. In case of an object recognition network, for example, the network might start learning edges, then corners, to more complicated patterns and finally the object [26].

CNN Structure: A CNN is a Deep Neural Network that usually consists of Convolutional layers, usually followed by an activation function like Rectified Linear Units (ReLU), Pooling layers and Fully-Connected layers. An example of a CNN is referenced in Figure 3.4.
CNN Algorithm: The Neural Network is essentially a function that maps the input to the output. So, the learning algorithm aims to minimize the training error between the predicted values of the network and the target values by iteratively updating the parameters of the network through backropagation of the error, which are quantified by the loss function.

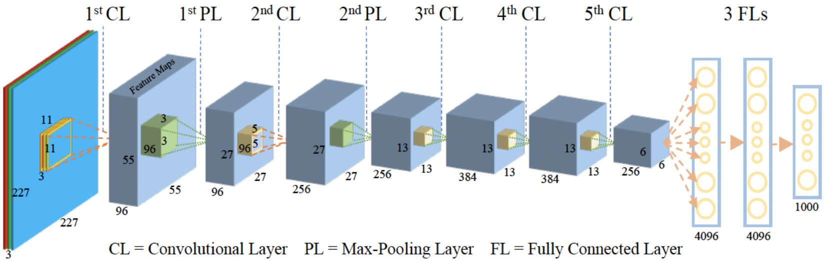


Figure 3.4: Internal working of a basic Convolution Neural Network. Image credits: [42].

3.7.2 OpenPose

We use the architecture of OpenPose, illustrated in Figure 3.5 through modifying the publicly available codebase [[60], [24]]. The network is derived from the Convolutional Pose Machines [59] architecture. The original OpenPose [7] architecture is a six-stage network. The first stage is a feedforward network which output a set of Part Affinity Fields (PAFs) and a set of belief maps. Belief map measures confidence of an image pixel to be a certain part. PAFs encode the relationship between two features or parameters of the object. In subsequent stages, the predictions from the previous stage and image features are used to refine predictions. In our training pipeline, the first stage outputs only a set of belief maps, instead of both belief maps and PAFs, and consists of 3 stages, instead of 6.

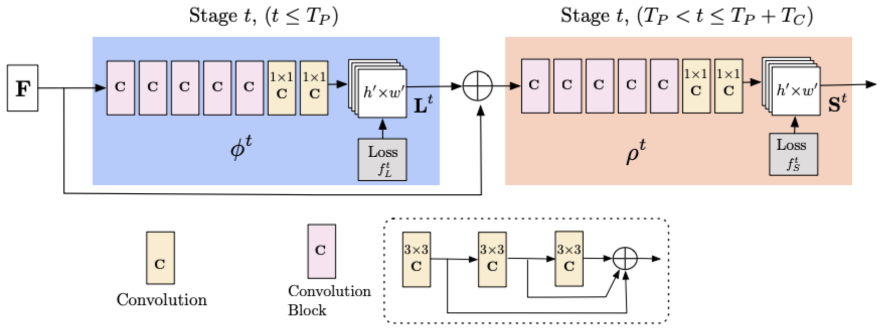


Figure 3.5: The architecture of OpenPose. Image credits: [7].

3.7.3 Convolutional Pose Machines

Convolutional Pose Machines [59] predict the pose of the hand in the RGB images. CPMs consist of a sequence of Convolutional Neural Networks. The output of each of these networks is a belief map that outputs the measure of belief (or probability) of each pixel belonging to each of the joints. This belief map is passed as input to the next network in the sequence. Each network refines the prediction from the previous and also consolidates these predictions with respect to the larger image sample. Thus, it preserves large-scale spatial dependencies. The architecture is graphically explained in Figure 3.6.

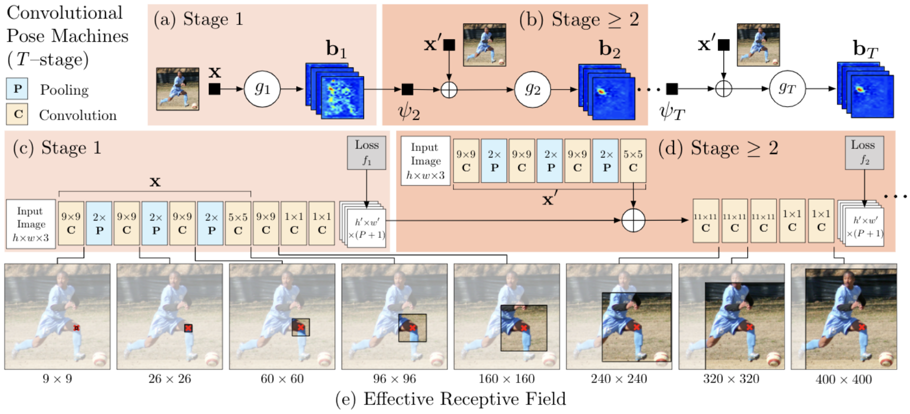


Figure 3.6: The architecture and effective layer-wise receptive field development in CPM. Image credits: [59].

3.8 Performance Inference Metrics

We use the following metrics for performance evaluation in this work:

1. 2D Keypoint Error (2DKPE): The average of the euclidian distance between the predicted and ground truth values per joint per images over test datasets as a standard metric in the works of [67, 18].

2. Area under the 2D curve (AU2D): The area under the curve for the percentage of keypoints that are below 2DKPE thresholds. From our experiments, in our test datasets of size 640×480px, 100px is a sufficiently high error and beyond this number, predictions were considered unsuccessful. Therefore, we set the upper limit to 100px.

4 Methodology

In the following chapter, we present our methods and motivations in detail. Section 4.1 provides an overview of our approach. Section 4.2 describes the data generation process. Section 4.3 illustrates the properties of the datasets used as training inputs alongwith image samples. Section 4.4 explains the training and inference process.

4.1 Overview of our approach

We provide a brief overview of the data generation and training part of the process in Figure 4.1 and and the evaluation process in Figure 4.2. The first step is to generate a suitable synthetic dataset using the MANO model [45] by modifying the accompanying codebase. This dataset may undergo further post-processing to add effects such as texture, noise, blur using the OpenCV library [6]. The second step is to train the model using OpenPose [7] with the dataset generated in the previous step as its input. Once the model is trained, we evaluate the quality of the model on real datasets. This process is typically twofold - a hand detector which provides a bounding box and a pose estimator (OpenPose), which takes the bounding box from the hand detector and outputs the predicted Hand Pose. In our method, we focus only on the problem of Pose Estimation and use a ground truth-based bounding box as the input to the Pose Estimation network.

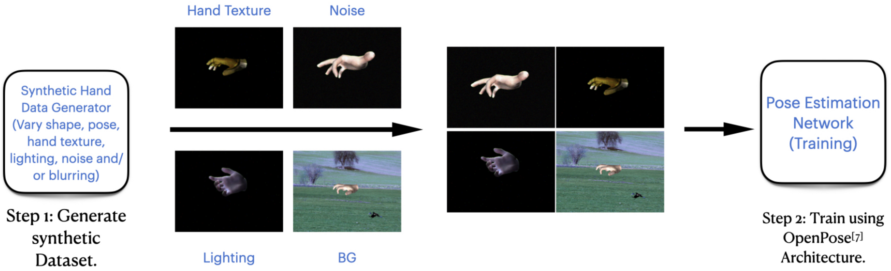

Synthetic Hand Data Generator (Vary shape, pose, hand texture, lighting, noise and/ or blurring)
Step 1: Generate synthetic Dataset.
Hand Texture
Noise
Lighting
BG
Pose Estimation Network (Training)
Step 2: Train using OpenPose[7] Architecture.

Figure 4.1: Overview of the Data Generation and training steps our approach.

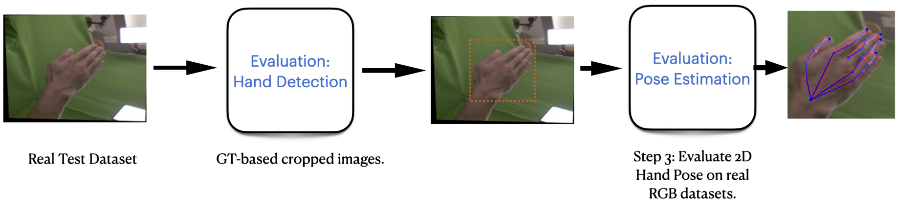

Evaluation: Hand Detection
Evaluation: Pose Estimation
Real Test Dataset
GT-based cropped images.
Step 3: Evaluate 2D Hand Pose on real RGB datasets.

Figure 4.2: Overview of the evaluation step our approach.

4.2 Generation of our synthetic datasets

4.2.1 Overview

We generate datasets by varying the parameters of the MANO model [45]. For this purpose, we modified the accompanying codebase and built directly on top of it. MANO is a statistical hand model. We vary the pose and shape parameters of the model to retrieve a mesh of varying hand shapes and articulation. We also use a rendering library called OpenDR [32] to render the hand mesh into a scene to add background, camera, lighting, mesh texture and positioning; and finally converting them to an RGB or depth image. The MANO Model has 45 DoFs for the 15 finger joints including the wrist (for the three axes of rotation), 6 DoFs for global rotation and position and 10 DoFs for shape of the hand. We load a model using 12 PCA parameters. These PCA parameters linearly regress the 45 hand pose parameters for the finger joints and wrist. We vary them randomly between -2 and +2 since this variance from the value of 0, i.e, the mean Hand Pose (shown in Figure 4.3) covers nearly all the feasible poses. This range of values are a result of experimentation where we generated poses to find a good trade-off between variability and feasibility. We run these iterations to generate a fairly comprehensive distribution of poses.

Note: We do not vary the global position of the hand. Instead, we randomly rotate the camera around the hand to be able to capture all viewpoint angles.

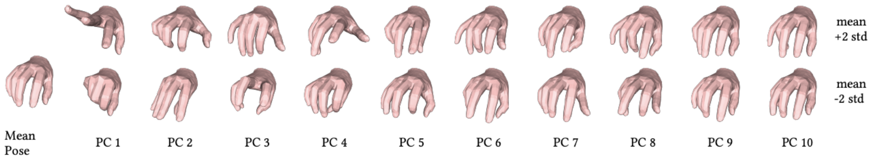


Figure 4.3: The mean pose (represented by the value of PCA parameters being 0) followed by the effect of varying the first 10 PCA parameters of the MANO model. Image credits: [45].

4.2.2 Types of Datasets

We generate the following types of datasets:

1. Base Datasets: containing 50-120K images of hand with skin coloured hand on a black background under good illumination conditions, with or without shape or lighting variations as detailed in Table 4.1. We vary the pose and shape parameters to cover nearly all feasible poses and use an articulation constraint to reject infeasible poses while allowing most feasible poses using angle limits of fingers from [56], displayed also in Table 4.2.

Label	Training Images	Shape parameters	Lighting
Base Dataset w/out shape and lighting (BD)	48K	Not varied	Not varied
Base Dataset w/ shape, w/out lighting (BD-S)	110K	Varied	Not varied
Base Dataset w/ shape and lighting (BD-SL)	110K	Varied	Varied

Table 4.1: Properties of the Base Datasets used in experiments.

Finger	**Distal IP** (°)	**Proximal IP** (°)	**metacarpophalangeal** (°)
Little Finger	-30 $\geq \theta \geq$ 90	0 $\geq \theta \geq$ 90	-10 $\geq \theta \geq$ 40
Ring Finger	-30 $\geq \theta \geq$ 90	0 $\geq \theta \geq$ 90	-10 $\geq \theta \geq$ 20
Middle Finger	-30 $\geq \theta \geq$ 90	0 $\geq \theta \geq$ 90	-15 $\geq \theta \geq$ 15
Index Finger	-30 $\geq \theta \geq$ 90	0 $\geq \theta \geq$ 90	-20 $\geq \theta \geq$ 10
Thumb	-10 $\geq \theta \geq$ 90	0 $\geq \theta \geq$ 90	-10 $\geq \theta \geq$ 100

Table 4.2: Articulation constraint for each finger (in degrees) [56].

2. Datasets with variance added to Base Dataset: There are many avenues of introducing randomness in the image of an articulated object. We choose to add variance in the background, hand texture, noise and lighting. We also use blurring. The variance details of each type of randomization parameter is mentioned in Table 4.3 and samples are provided in Figure 4.4.

Variance Parameter	**Type**
Background	Black, Uniform, ImageNet Images [10]
Hand Texture	Skin colour, Randomly sampled vertex colours from ImageNet image, ImageNet images as textures
Noise	Compression artifacts, Salt and Pepper Noise, Gaussian Noise
Lighting	Lambertian white light, Randomly coloured Lambertian light
Blurring	Gaussian Blur

Table 4.3: Variance parameter details used in dataset generation.

a) The texture that is applied on the hand may be of the following types. We provide samples in Figure 4.4 (a):

i. Skin colour.

ii. Randomly sampling of points on random ImageNet [10] images.

iii. Wrapping randomly sampled ImageNet images over the hand, in post-processing. Sometimes, this leads to extremely dark images.

To prevent this, we restrict the RGB values ≥ 128 and apply histogram equalisation.

b) The background we apply to the image may be of the following types. We provide samples in Figure 4.4 (b)):

   i. Uniform black background.
   ii. Randomly sampled ImageNet images that are resized to the image size, i.e, 640×480px.

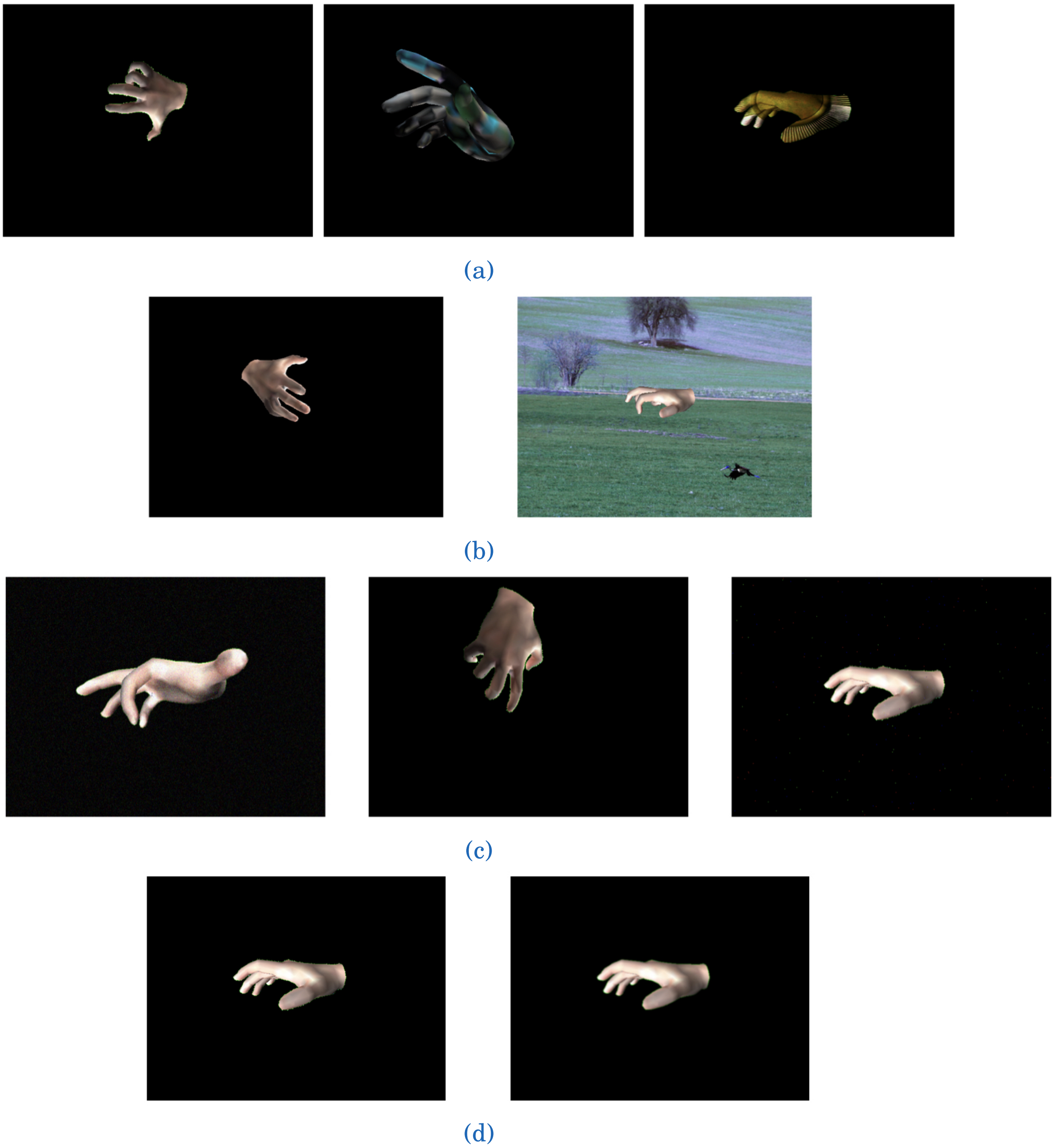
c) The noise we introduce to the image may be of the following types We provide samples in Figure 4.4 (c):

   i. Salt and Pepper Noise: Using OpenCV [6], we add it to 0.5 percent of the pixels.
   ii. Compression artifacts: We introduce this through the segmentation of the hand from its background using jpeg or other noisy png segmentation masks.
   iii. Gaussian Noise: Using OpenCV [6], we add Gaussian Noise of mean 0 and variance 0.005.

d) We randomize the lighting in the scene by adding randomly coloured Lambertian light sources in the renderer.

3. Mixed datasets: We have three types of mixed datasets.

   a) Datasets combining images from datasets with variance in a single, unique parameter without lighting or shape variations Figure 4.15 (a) or with shape and/or lighting variations.
   b) Datasets combining images with incremental variance in multiple parameters without lighting or shape variations Figure 4.15 (b) or with shape and/or lighting variations. In Figure 4.15 (d), random lighting has been added.
   c) Datasets containing poses from base datasets with multiple backgrounds or hand textures Figure 4.15 (c).



(a)

(b)

(c)

(d)

Figure 4.4: Details of variance applied to the base dataset: (a) Variance in the hand textures. (b) Variance in the background. (c) Variance of the noise and (d) Application of Gaussian Blur.

4.3 Details of the trained models

The following section describes in detail the properties of the images in our synthetic datasets. We also provide image samples corresponding to their training dataset.

4.3.1 Dataset Details

The parameter variation of generation of datasets for the purposes of training have been delineated in Table 4.4 for single parameter variance, Table 4.5 for incremental parameter variance and Table 4.6 for mixed datasets formed by combining some of the individual datasets. Please note, as mentioned in Table 4.1, we also have base datasets with shape and/or lighting variations with the parameter variations that we apply random parameter variations on top of. In the following section, we take the base dataset without shape or lighting variation as an example to present the properties of datasets.

Label	Training Images	Hand Texture	Image BG	Noise Introduced	Blur
Base Dataset w/out shape and lighting (BD)*	48K	Skin colour	Black	JPEG (Compression Artifact)	None
BD + Random Hand Texture	48K	**Random Texture (ImageNet)**	Black	JPEG (Compression Artifact)	None
BD + Random BG	48K	Skin Colour	**Random BG (ImageNet)**	JPEG (Compression Artifact)	None
BD + Gaussian Noise	48K	Skin Colour	Black	**Gaussian (mean=0, var=0.005)**, JPEG (Compression Artifact)	None
BD + Gaussian Blur	48K	Skin Colour	Black	JPEG (Compression Artifact)	**Gaussian (Window=5,5)**
BD + S&P Noise	48K	Skin Colour	Black	**Salt and Pepper (amount=0.005 percent)**, JPEG (Compression Artifact)	None

Table 4.4: Dataset parameter details for individual parameter variance on top of the base dataset without shape and lighting variations.

4.3.2 Base Dataset without Shape or Lighting Variations (BD)

In the following Figure 4.5, we provide samples of the training dataset without Domain Randomization (except for pose randomization). We use this to calculate the benchmark, as a base dataset that we apply variance on and as the first stage in our finetuning pipeline. This base dataset does not contain any shape or lighting variations (BD). But, we have other base datasets with variations in shape (BD-S) and, in both shape and lighting (BD-SL), as detailed above in Table 4.1.

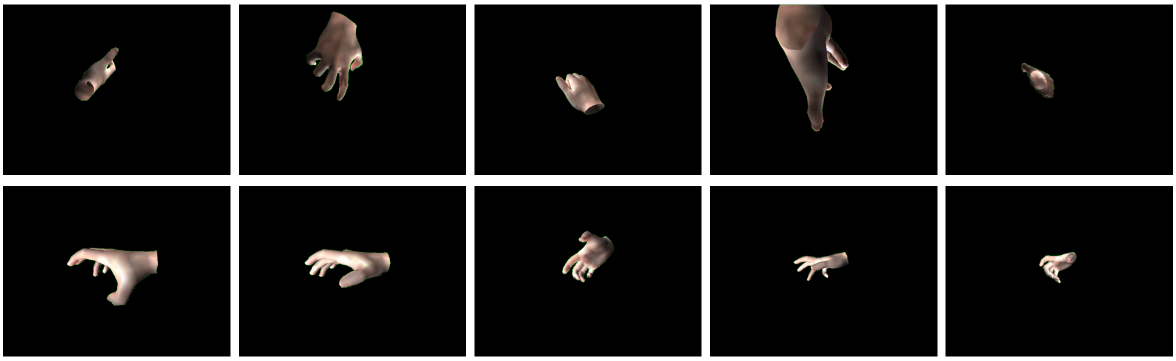


Figure 4.5: Training image samples of the base dataset (without shape and lighting variations).

4.3.3 Datasets Containing Variance in a single parameter

In the following section, samples from datasets containing randomization of single parameters on top of the base dataset (BD) have been displayed. This includes adding variance to the image background, hand texture, shape, lighting or addition of Gaussian or S&P Noise or Gaussian Blur.

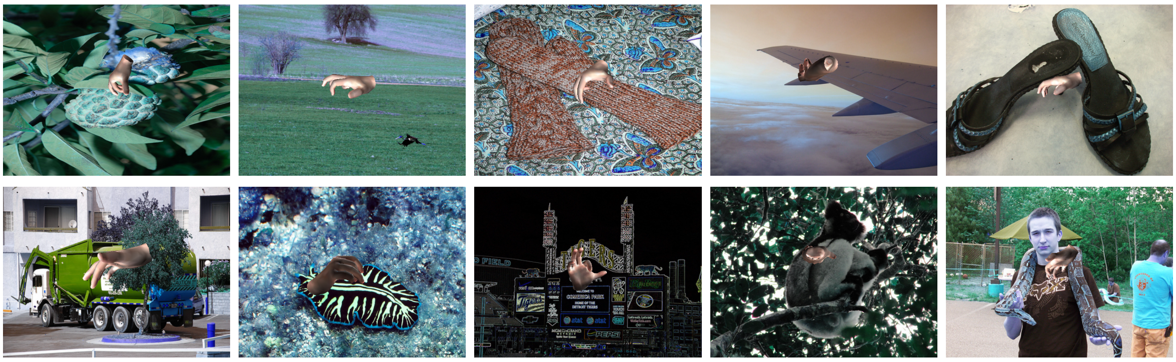


Figure 4.6: Training image samples of dataset with addition of a Random BG image on top of the base dataset (BD).

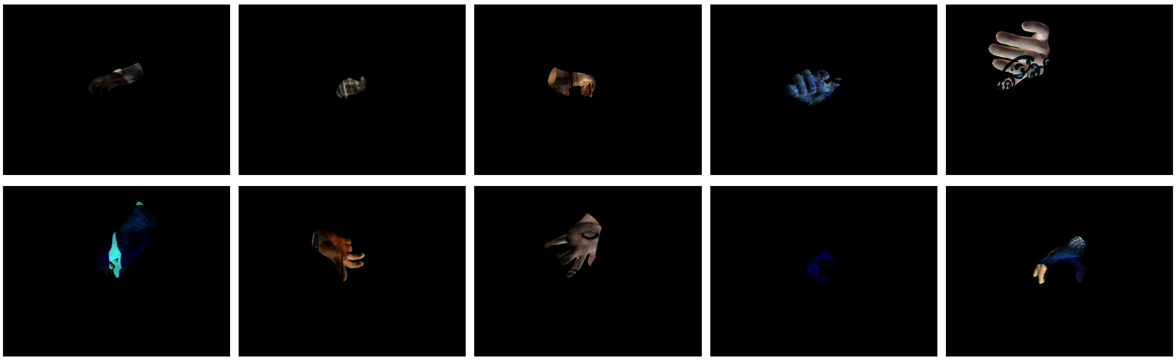


Figure 4.7: Training image samples of dataset with addition of a Random Hand Texture on top of the base dataset (BD).

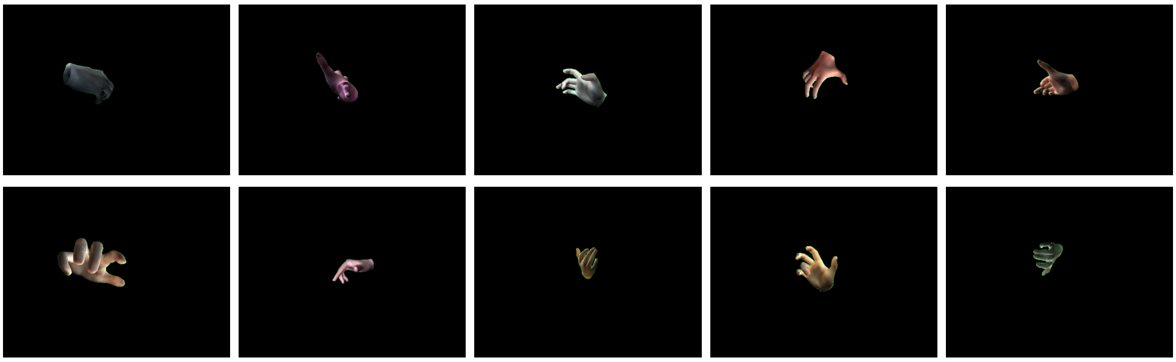


Figure 4.8: Training image samples of dataset with addition of random lighting on top of the base dataset.

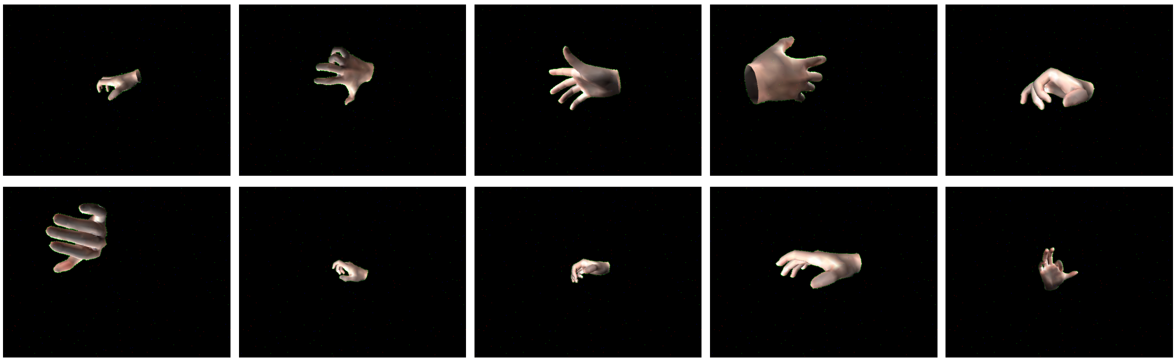


Figure 4.9: Training image samples of dataset with the addition of Salt and Pepper Noise on top of the base dataset (BD).

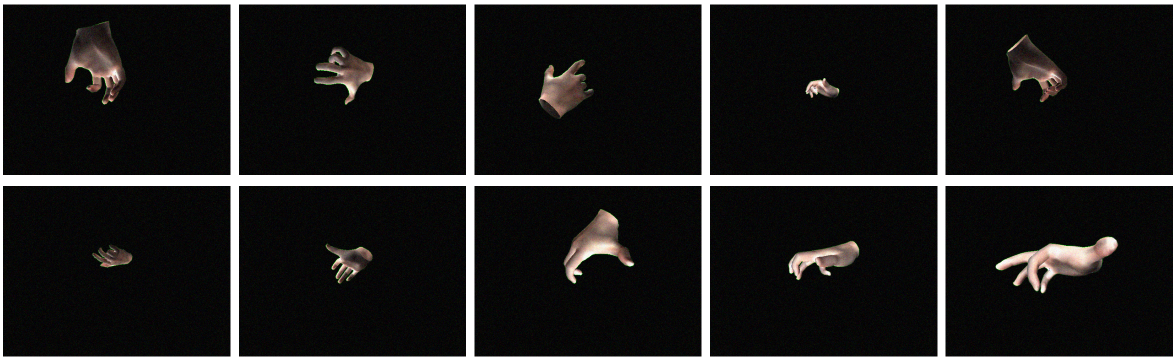


Figure 4.10: Training image samples of dataset with the addition of Gaussian Noise on top of the base dataset (BD).

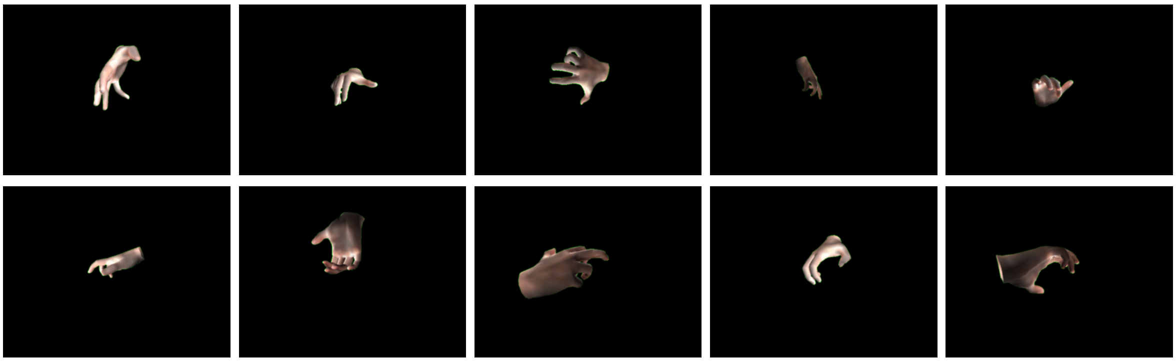


Figure 4.11: Training image samples of dataset with the addition of Gaussian Blur on top of the base dataset (BD).

4.3.4 Datasets Containing Variance in multiple parameters

In the following section, samples from datasets containing randomization of multiple parameters on top of the base dataset (BD) have been displayed. We add this randomization incrementally. The most notable progression of parameter variance has been tabulated in Table 4.5. So, on top of the base datasets, we can add Random Hand Texture and Random BG. Further, on top of this, we can add S&P Noise, and finally, on top of that, we can add Gaussian Blur.

Properties	Training Images	Hand Texture	Image BG	Noise Introduced	Blur
BD + Random BG	48K	Skin colour	**Random BG (ImageNet)**	JPEG (Compression Artifact)	None
BD + Random Hand Texture	48K	**Random Texture (ImageNet)**	None	JPEG (Compression Artifact)	None
Previous + Random BG	48K	**Random Texture (ImageNet)**	**Random BG (ImageNet)**	JPEG (Compression Artifact)	None
Previous + S&P Noise	48K	Random Texture (ImageNet)	Random BG (ImageNet)	**Salt& Pepper (amount=0.005 percent)**, JPEG (Compression Artifact)	None
Previous + Gaussian Blur	48K	Random Texture (ImageNet)	Random BG (ImageNet)	Salt and Pepper (amount=0.005 percent) , JPEG (Compression Artifact)	**Gaussian (Window=5,5)**

Table 4.5: Dataset parameter details for incremental parameter variance on top of the base dataset (BD).

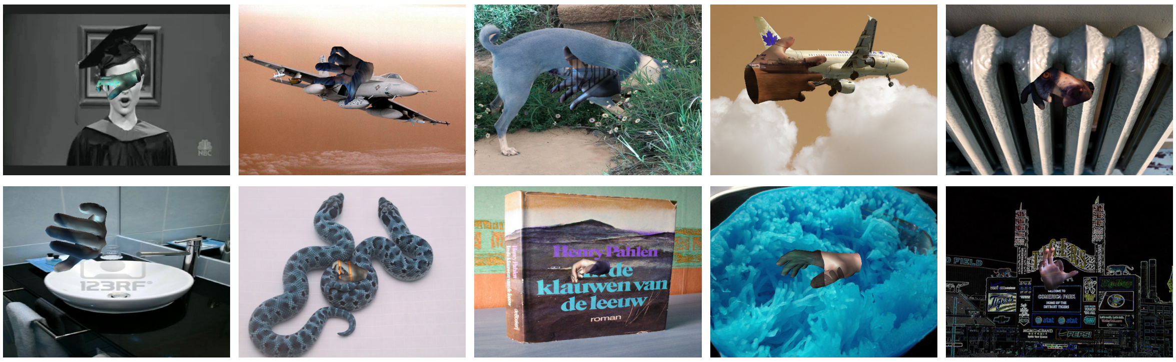


Figure 4.12: Training image samples of dataset with the addition of Random Hand Texture and Random BG on top of the base dataset (BD).

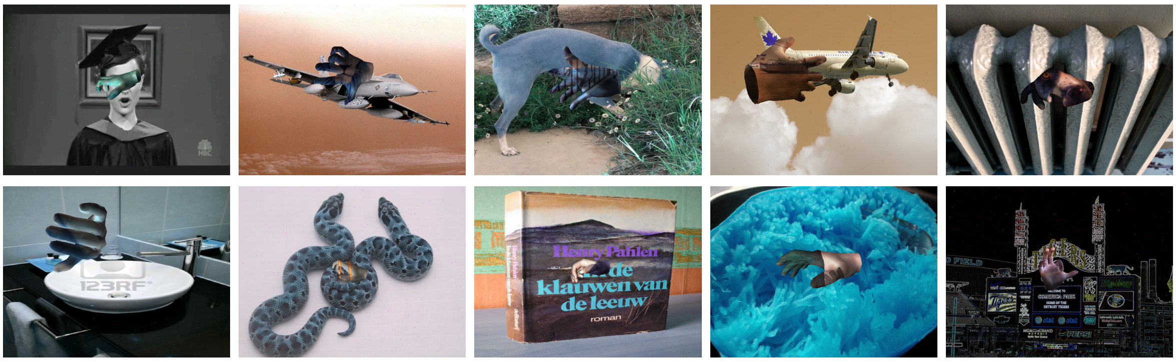


Figure 4.13: Training image samples of dataset with the addition of Random Hand Texture, Random BG and Salt and Pepper Noise on top of the base dataset (BD).

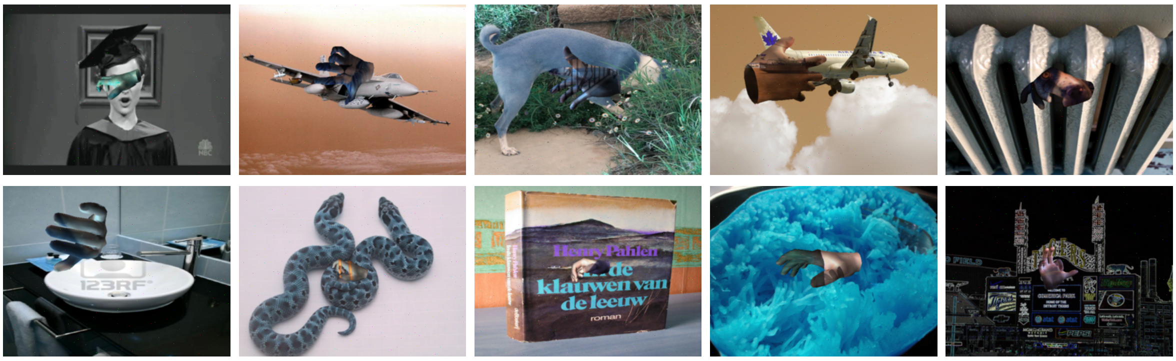


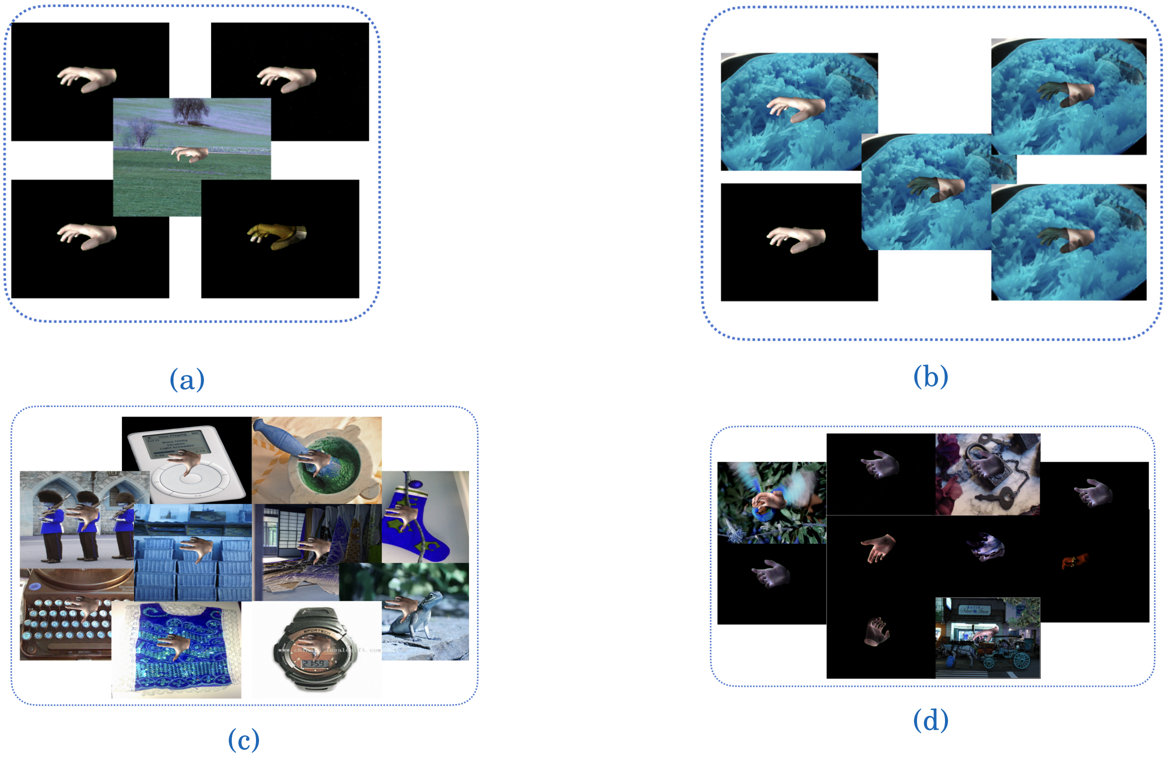
Figure 4.14: Training image samples of dataset with the addition of Random Hand Texture, Random BG, Salt and Pepper Noise and Gaussian Blur on top of the base dataset (BD).

4.3.5 Mixed Datasets

By combining the datasets mentioned in the above sections containing variance in single or multiple parameters, we generate mixed datasets. Some notable datasets have been described in Table 4.6. Some samples from these datasets have been displayed in Figure 4.14.

Properties	Training Images	Description
Mixed individual variance **(w/out shape and lighting)**	240K	A combination of the base dataset w/out shape, lighting variation with four datasets containing randomization of single parameters (Random BG, Random Hand Texture, Salt and Pepper Noise, Gaussian Blur) .
Mixed incremental variance **(w/out shape and lighting)**	240K	A combination of the base dataset w/out shape, lighting variation with four datasets containing incremental randomization of parameters in the following order: Random BG, Random Hand Texture, Salt and Pepper Noise and Gaussian Blur.
Mixed multiple BG **(w/out shape and lighting)**	480K	Each pose from the base dataset w/out shape, lighting variation repeated in front of 10 randomly sampled ImageNet backgrounds.
Mixed individual variance **(w/ shape and lighting)**	240K	A combination of the base dataset w/ shape, lighting variation with four datasets containing randomization of single parameters (Random BG, Random Hand Texture, Salt and Pepper Noise, Gaussian Blur) datasets.
Mixed individual variance **(w/ shape w/out lighting)**	545K	Similar mixture as row one with randomized shape parameter and increased number of images.
Mixed incremental variance **(w/ shape w/out lighting)**	545K	Similar mixture as row two with randomized shape parameter and increased number of images.

Table 4.6: Mixed Datasets.



(a)

(b)

(c)

(d)

Figure 4.15: Mixed Datasets containing a (a) Mixed Dataset containing a combination of five datasets, i.e, single parameter variance of Random BG (randomly sampled from ImageNet), Random Hand Texture (randomly sampled from ImageNet), Salt and Pepper noise and Gaussian Blur on top of a base dataset without shape or lighting variation, along with the base dataset Mixed individual variance (without shape and lighting). (b) Mixed Dataset containing a combination of five datasets with incremental parameter variance on top of a base dataset without shape or lighting variation in the following order: Random BG, Random Hand Texture, Salt and Pepper noise and Gaussian Blur, along with the base dataset Mixed incremental variance (without shape and lighting). (c) combination of ten randomly sampled backgrounds for the same pose Mixed multiple BG (without shape and lighting) (d) Mixed Dataset containing a combination of five datasets, i.e, single parameter variance of Random BG (randomly sampled from ImageNet), Random Hand Texture (randomly sampled from ImageNet), Salt and Pepper noise and Gaussian Blur on top of a base dataset with lighting and shape variation, along with the base dataset Mixed individual variance (with shape and lighting).

4.3.6 Ground Truth Generation

To generate the ground truth labels, we retrieve 21 3D locations of finger joints including the wrist. The original MANO model only returns 16 joints (excluding the fingertips). So, for our experiments, we manually select vertices with the indices of the fingertips 745, 320, 444, 555, 657 from the generated mesh of the hand. We used the Open3D library [65] to visualise and locate the vertices of the fingertips on the mesh. These 3D locations were subsequently projected into the 2D image coordinate system.

Parameter	Value
Image input shape	640×480
Input ground truth data	21×2 (x, y coordinates of each joint)
Number of OpenPose stages	3
Cropped image size	300×300px
Rotational augmentation	Varies from [-180, 180]
Scale augmentation	Varies from [0.6, 1.4] times the original image
Library used	tensorflow v1.10
GPU	NVIDIA
Batch size	8

Table 4.7: Parameters related to the training pipeline.

4.4 Training and Inference Process

4.4.1 Network Architecture

The network architecture used for the purpose of training is from the OpenPose paper [7]. We modified the publicly available codebase *MonocularTotalCapture* [60] for the implementation of the training architecture. As mentioned in Section 3.8, OpenPose is based on the CPM architecture. We used this network with 3 stages to predict the 2-D locations of the 21 hand joint locations displayed in Figure 3.2.

It is a state-of-the-art architecture for the training and prediction of 2-D locations of hand joints from an RGB image in a frame-by-frame, non-tracking context. The pose prediction networks that output a per-pixel, per-joint confidence map like CPM instead of regressing to 21 parameters generally have a better performance.

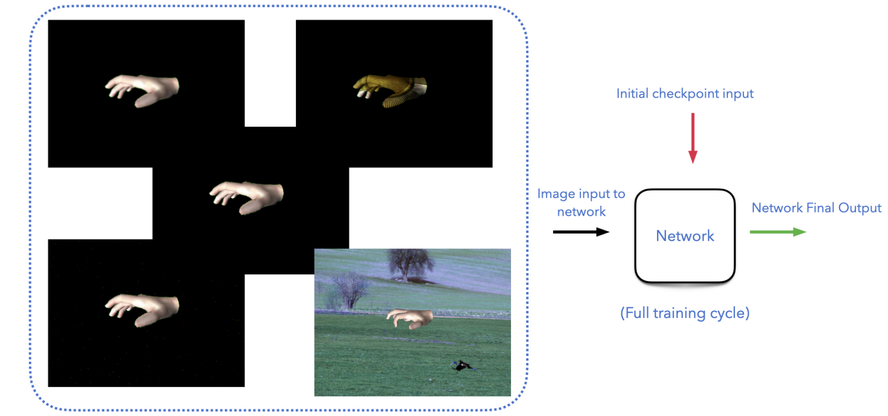
4.4.2 Training process pipeline

We performed two types of training:

1. Training on individual datasets only until convergence, which takes approximately 350K iterations. We did this for base and mixed datasets.
2. Training in multiple stages (finetuning): Training until convergence on one dataset for approximately 350K iterations and finetuning on the same network in subsequent stages using incremental addition of variance to

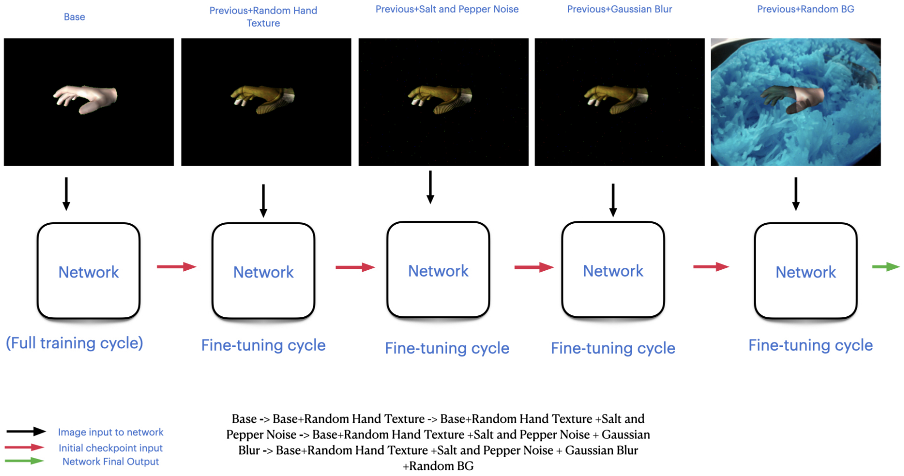
images in datasets from the previous stage to enhance the prediction for approximately 3K iterations. The reasons behind using finetuning was to:

- Avoid forgetting in the network.
- Training by introducing randomization of multiple parameters in the same image led to suboptimal performance during evaluation.
- Conduct ablation studies and,
- To choose effective parameters to form mixed datasets.



Mixed Dataset containing approximately 50K images each of base dataset, base+random hand texture, base+random BG, base+salt and pepper noise and base+gaussian blur.

Figure 4.16: Full Training on a mixed dataset.



Base -> Base+Random Hand Texture -> Base+Random Hand Texture +Salt and Pepper Noise -> Base+Random Hand Texture +Salt and Pepper Noise + Gaussian Blur -> Base+Random Hand Texture +Salt and Pepper Noise + Gaussian Blur +Random BG

Figure 4.17: Finetuning in multiple stages through incremental variance in parameters.

4.4.3 Inference Process

We use checkpoints from our trained model to predict the locations of the 21 salient joints of the hand on images from real world test datasets of size

640×480px. We first generate a bounding box of size 300×300px after using the center of the palm and the finger extremities from using the ground truth labels. Next, we pass this cropped image through our pose estimation network which outputs 21×2 locations of the joints. Finally, we compare the predictions with the existing ground truth data to generate performance metrics like the 2DKPE and AU2D as described in Section 3.9.

5 Evaluation

In the following chapter, we will provide details of our evaluation methods, the test datasets used, results and observations.

Section 5.1 explains the testing environment. Section 5.2 details the properties of the test datasets used for evaluation of our trained models. Section 5.3 provides details of the evaluation results of the experiments. Section 5.4 draws a comparison between the results obtained without Domain Randomization as a benchmark.

5.1 Testing Environment

The input sizes of all the images are 640×480px. We focus solely on the task of Hand Pose Estimation and assume a bounding box of size 300×300px according to the extremities of the hand from the first image of the sequence from the ground truth data. The assumption is made on the basis that our network could be attached to a state-of-the-art Hand Detection Network to procure a bounding box which can be used for the purpose of Hand Pose Estimation. Here, we choose to circumvent that step by assuming a bounding box.

5.2 Test Datasets description

Test datasets contain real RGB images, originally or resized (in case of Freihands datasets) to 640×480px containing hands without objects with ground truth consisting of either 3D or 2D locations of 20 or 21 joints. Since our network focuses on 2D pose estimation, we projected 3D ground truth data to the 2D image coordinate system. Table 5.1 tabulates the salient properties of all our test datasets.

Test Dataset	Year	Images Sampled/Total	Joints	Background
Freihands Training	2019	2721/130K	21	Real, Green room
Freihands Training w/o BG	2019	788/130K	21	Segmented, Black
STB - Random gestures	2017	9K/9K	21 (without wrist)*	Real, 6 backgrounds
STB - Counting gestures	2017	9K/9K	21 (without wrist)*	Real, 6 backgrounds
Large-scale Multiview 3D Hand Pose Dataset	2017	10K/80K	21 (without wrist)*	Real
Large-scale Multiview 3D Hand Pose Dataset with Augmentation	2017	10K/80K	21 (without wrist)*	Augmented
Sampled from HO-3D dataset	2019	327	21	Real

Table 5.1: Properties of the real world test datasets. *indicates the ground truth labels of these datasets (STB and Large-scale Multiview 3D Hand Pose datasets) contain the location of the palm center instead of the wrist. Therefore, we only compare the locations of 20 joints (excluding the wrist) for our error calculation.

Freihands Training Dataset:
We randomly sampled around 3K images from the training dataset. This dataset contains hands of multiple people in front of either a uniform green or a cluttered background. The hand is well illuminated in the images. We provide some samples of the dataset in Figure 5.1.

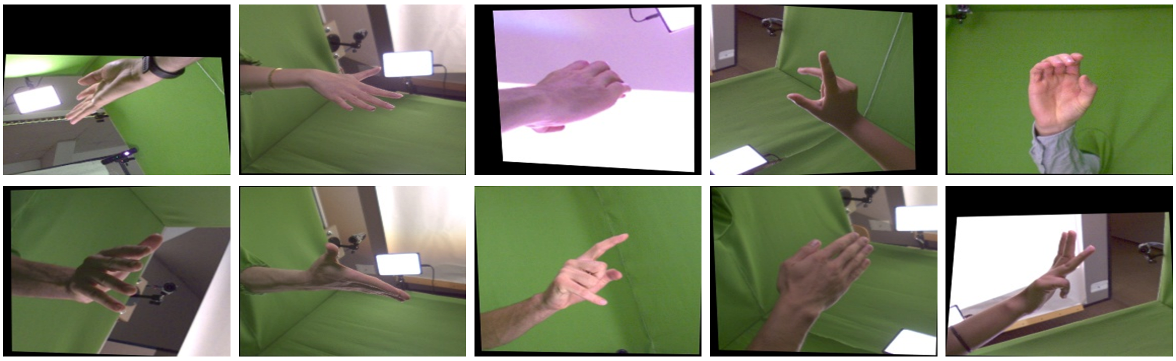


Figure 5.1: Image samples from Freihands Dataset from their training dataset [68].

Freihands Training Dataset without BG:
We sampled around 800 images from the Freihands training dataset, removed images with objects and segmented the hands from their backgrounds using their ground truth segmentation masks. We generated this dataset to provide a low-variance test dataset to ascertain the performances of models trained without randomization in many parameters, such as lighting and background. This made it possible for us to choose base datasets to incrementally add variance upon. We provide some samples of the dataset in Figure 5.2.

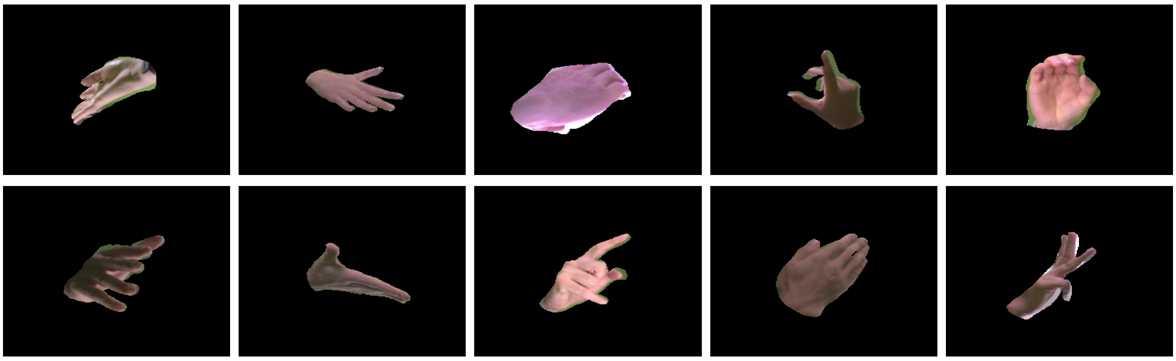


Figure 5.2: Image samples from the *Freihands Training Dataset* with the background removed [68].

STB Dataset containing Random gestures:
We used all 9K images of STB Dataset containing random gestures. They contain hand samples from a single person in front of 6 different backgrounds. We provide some samples of the dataset in Figure 5.3.

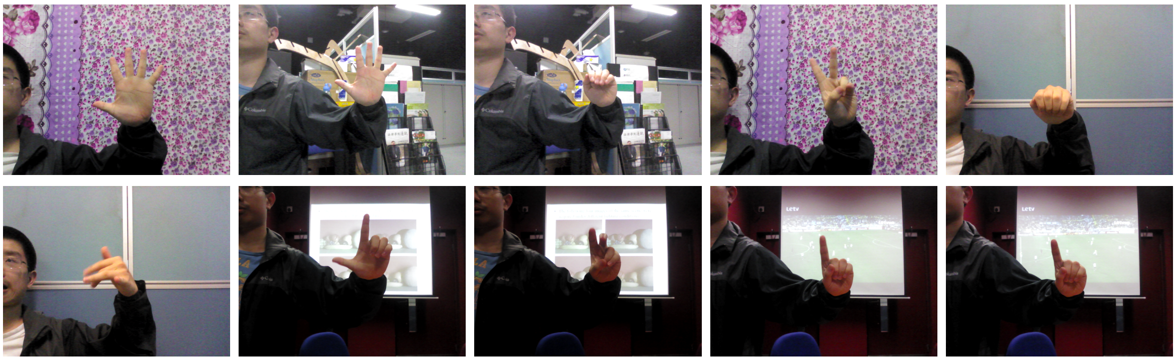


Figure 5.3: Image samples from the *STB Random Gestures Dataset* [64].

STB Dataset containing Counting gestures:
We used all 9K images of STB Dataset containing counting gestures. They contain hand samples from a single person in front of 6 different backgrounds. We provide some samples of the dataset in Figure 5.4.

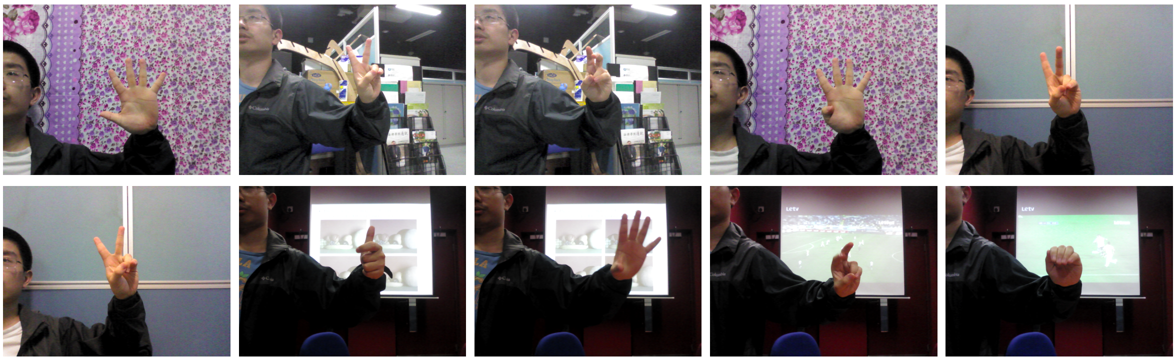


Figure 5.4: Image samples from the *STB Counting Gestures Dataset* [64].

Large-scale Multiview 3D Hand Pose Dataset:
We randomly sampled 10K images from the Large-scale Multiview 3D Hand Pose Dataset. It consists of 9 people in front of real, cluttered backgrounds. The motion of the hand was unrestricted. In two sequences, a subject wore different gloves, and in another sequence, a subject wore a mask. We provide some samples of the dataset in Figure 5.5.

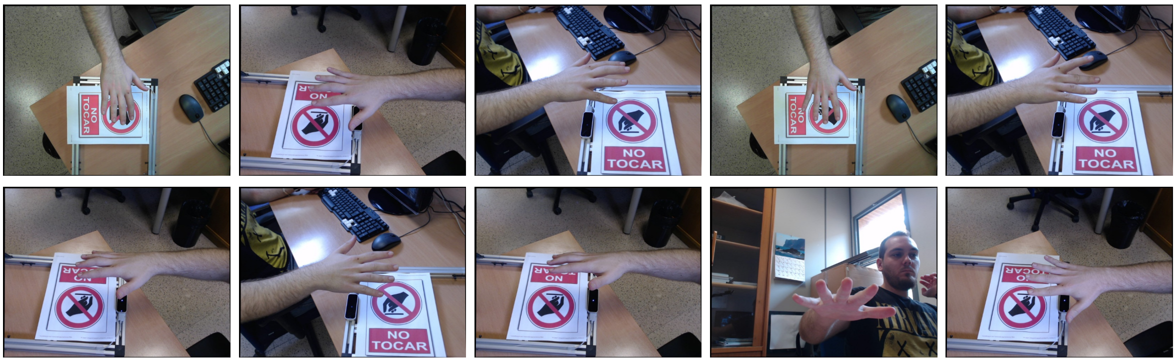


Figure 5.5: Image samples from the *Large-scale Multiview 3D Hand Pose Dataset* [16].

Large-scale Multiview 3D Hand Pose Dataset with background Augmentation:
We randomly sampled 10K images from the Large-scale Multiview 3D Hand Pose Dataset with background Augmentation. It consists of the augmentation of real hands captured in the previous dataset (Large-scale Multiview 3D Hand Pose Dataset) in front of cluttered backgrounds. It shares the same properties in terms of the hand. We provide some samples of the dataset in Figure 5.6.

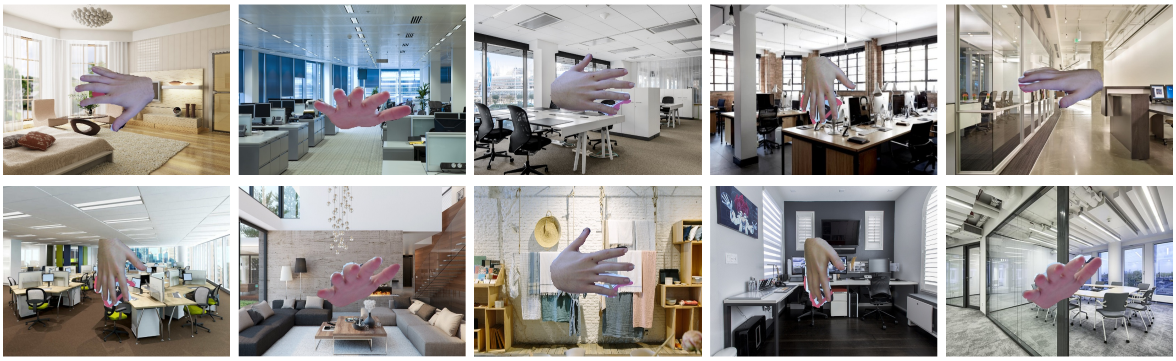


Figure 5.6: Image samples from *Large-scale Multiview 3D Hand Pose Dataset with background Augmentation* [16].

HO3D Dataset (Without objects):
We sampled approximately 300 images from a dataset captured in the same environment as the HO-3D Dataset without any object interactions. The geomteric pose of the hand is limited but this dataset provides variance in hand shape and skin colour.

We provide some samples of the dataset in Figure 5.7.

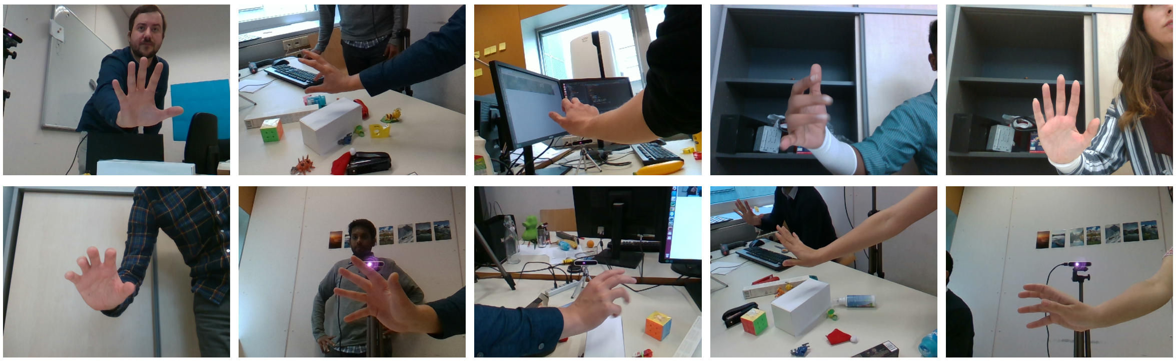


Figure 5.7: Image samples from images captured in the same setup as *HO-3D Dataset* without objects.

5.2.1 Data Distribution

We compute a t-SNE distribution of the 2D location of the joints (pose) to verify that:

1. We have a sufficiently wide variety of poses while training and testing.
2. There is enough overlap between the distribution of the test datasets and training datasets.
3. To ascertain the online scale+rotational augmentation required.

From Figure 5.8, we can observe that our training dataset covers most poses in the test data distribution of the Freihands dataset. We cover a lot of poses from the STB and MHP datasets. We also observe STB, MHP, Freihands Datasets cover a sufficiently large pose space and the HO-3D Dataset contains only a limited range of poses. This is because this is a primarily a hand+object dataset and we could not acquire many images without objects alongwith their ground truth labels.

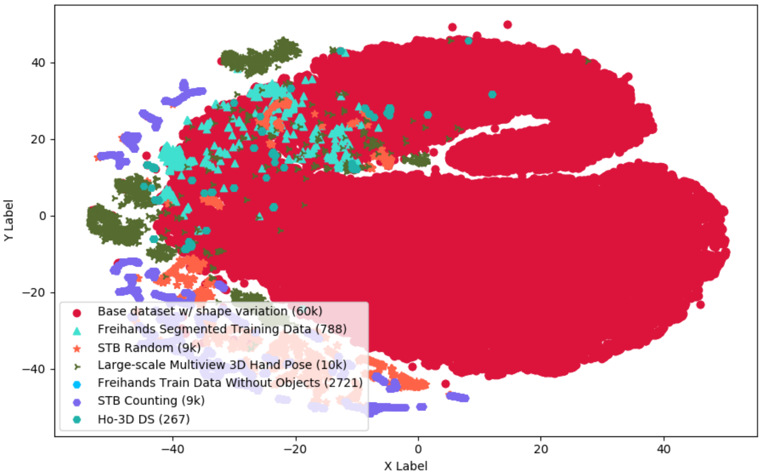


Figure 5.8: t-SNE distribution of the 2D location of the joints (pose) of all the test datasets with BD-S (Base Dataset with shape variation).

5.3 Evaluation results

In this section, we demonstrate the performances of the datasets in detail. Section 5.3.1 illustrates the error correspondences of various datasets through images. Section 5.3.2 provides details of the evaluation results for different trained models. Section 5.3.3 presents details of the ablation studies.

5.3.1 Error Metric Correspondences

The value of the 2DKPE depends not only on the performance of the network but also on the size of the hand in the image, accuracy of the ground truth and the size of the image. All our input images are cropped using a GT-based bounding box and are of the same size. So, the only differences in our test datasets are the sizes of the hand and the ground truth provided in the datasets. During training, we ensure that the sizes of the hands in the training datasets are in the range of that in our test datasets. In this section, we have provided examples of the correspondence of some error measurements (2DKPE) to visual error depiction through images for the different types of datasets, i.e, the Freihands Datasets in Figure 5.9, the STB Datasets Figure 5.10, Large-scale Multiview 3D Hand Pose images in Figure 5.11 and the HO-3D Dataset Figure 5.12. Please note that in the following figures, the blue represent the predictions of our model and the red lines represent the GT values.

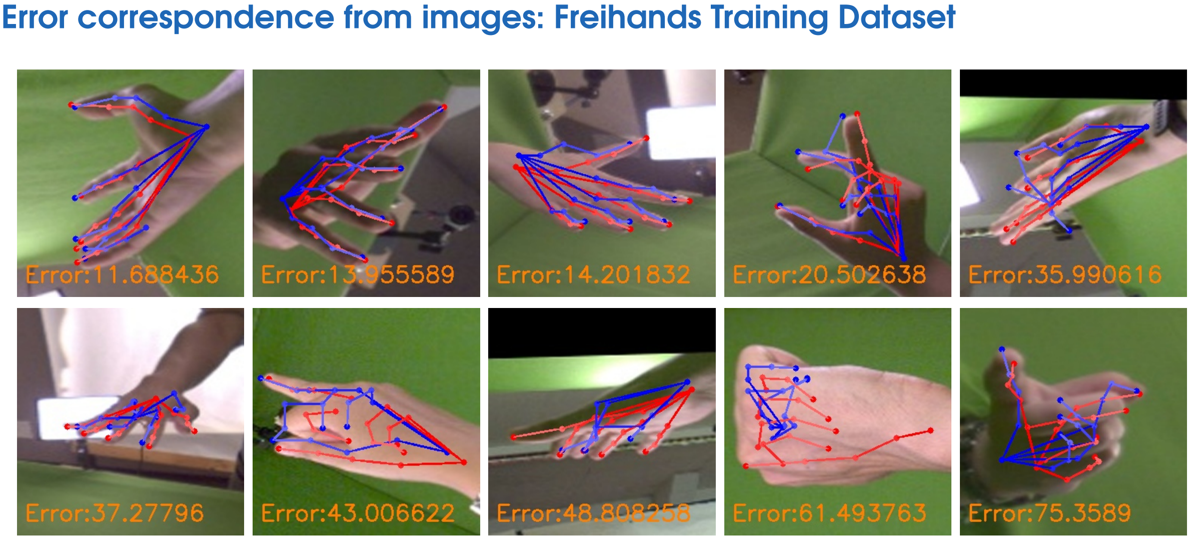

Error correspondence from images: Freihands Training Dataset

Figure 5.9: Error measurement (2DKPE) of predictions for the images in the Freihands Datasets.

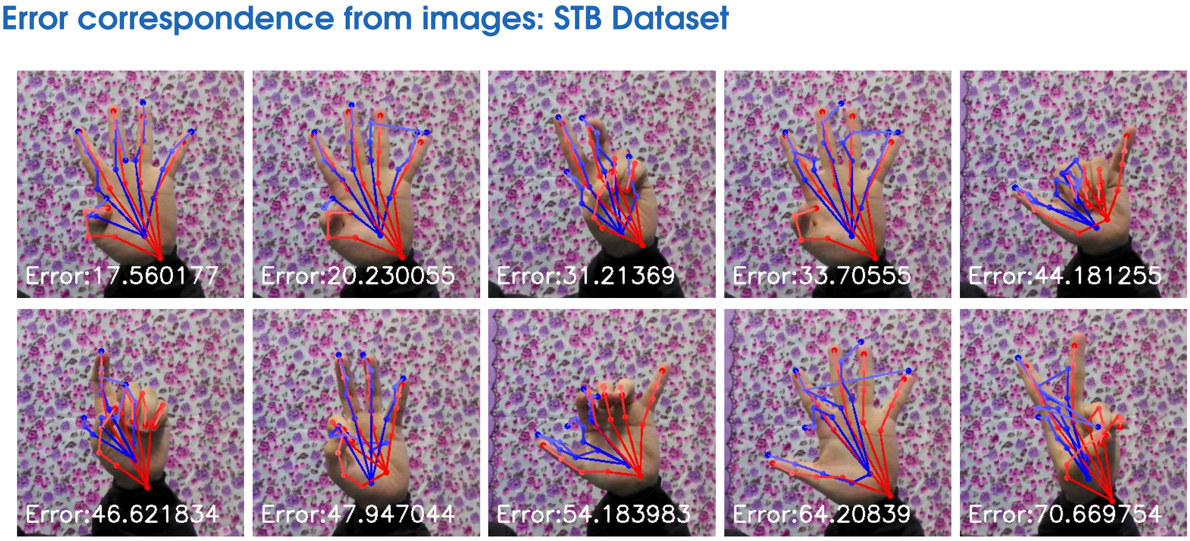

Error correspondence from images: STB Dataset

Figure 5.10: Error measurement (2DKPE) of predictions for the images in the STB Datasets.

Error correspondence from images: Large-scale Multiview 3D Hand Pose Dataset

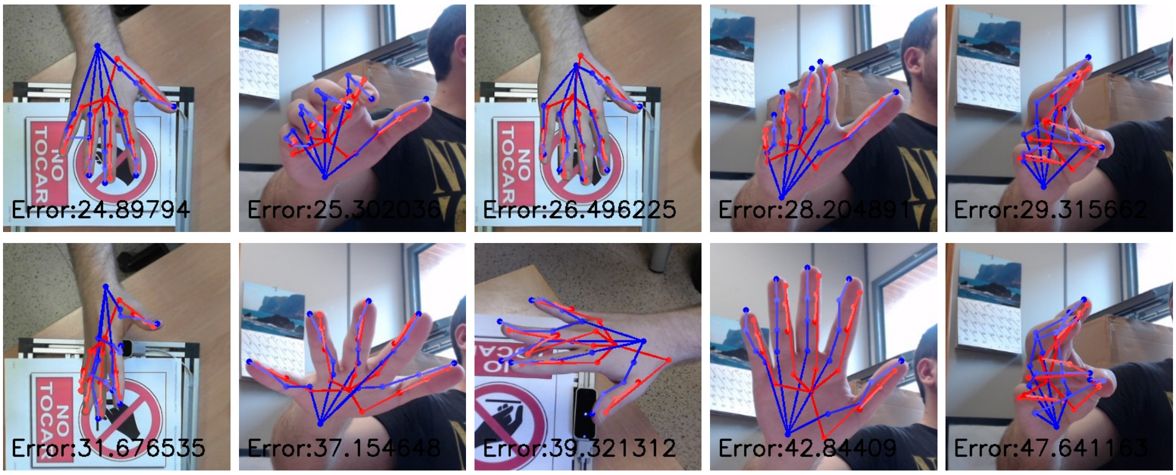


Figure 5.11: Error measurement (2DKPE) of predictions for the images in the Large-scale Multiview 3D Hand Pose Datasets.

Error correspondence from images: HO-3D Dataset

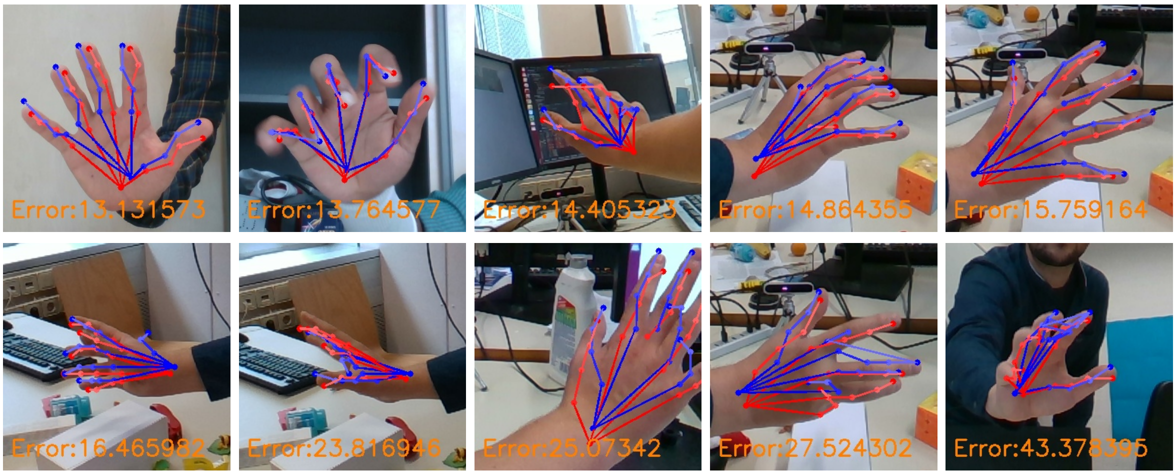


Figure 5.12: Error measurement (2DKPE) of predictions for the images in the HO-3D Dataset.

5.3.2 Overall Metrics

In this section, we discuss the performance of trained models using some mixed datasets and both types of training, i.e, finetuning and full training until convergence. Figure 5.13 plots the metrics for some of the most effective trained models on the real world datasets in accordance with Table 5.2. The randomization of pose, shape, lighting, hand texture and image background and the addition of Gaussian Blur contribute positively to the performance of the trained model. The test datasets have different properties as described in Table 5.1. Overall, 2DKPE of the model trained on mixed datasets containing datasets with incremental variance across STB Counting (40.28px) and STB Random (47.21px) Datasets, which have a high background variance, the models trained on datasets with variance in single/multiple parameters perform well. Compared to the highest 2DKPE amongst the training datasets, it reduces the error by 7px in STB Counting Dataset and 4px in STB Random Dataset. When compared to the performance of a model trained without Domain Randomization, it reduces the 2DKPE by half. Finetuning of the model incrementally on datasets with variance in the image BG and hand texture, without the addition of noise performs well on test datasets such as Freihand and Large-scale Multiview 3D Hand Pose (with and without BG augmentation). Compared to the highest 2DKPE amongst the training datasets, it reduces the error by 5px in Freihands Dataset, and the Large-scale Multiview 3D Hand Pose Datasets. When compared to the performance of a model trained without Domain Randomization, it reduces the 2DKPE by two-thirds (Freihands Dataset) and one-third (Large-scale Multiview 3D Hand Pose Datasets). As mentioned in the Table 5.2, the best performance on the Freihands test dataset with uniform black BG was achieved without randomization of the BG. Another observation from these figures is variance in the shape of the hand is effective.

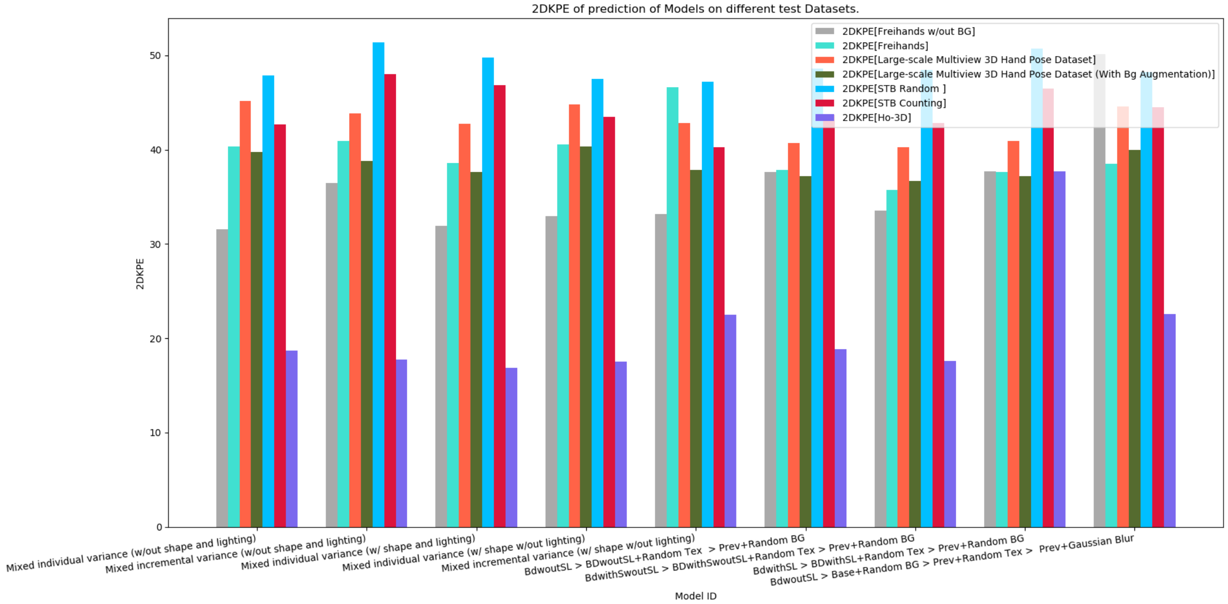


Figure 5.13: Comparison of models performances trained using mixed datasets with parameter variances or using our finetuning training pipeline using incremental parameter variances.

Model	Freihands	HO-3D	MHP	MHP (Aug)	STB (C)	STB (R)	Freihands w/out BG*
Mixed individual variance (w/out shape and lighting)	40.305	18.722	45.200	39.785	42.682	47.840	**31.552**
Mixed incremental variance (w/out shape and lighting)	40.960	17.758	43.868	38.786	47.983	51.377	36.477
Mixed individual variance (w/ shape and lighting)	38.608	**16.875**	42.746	37.644	46.825	49.773	31.946
Mixed individual variance (w/ shape w/out lighting)	40.551	17.515	44.832	40.351	43.504	47.537	32.984
Mixed incremental variance (w/ shape w/out lighting)	46.607	22.491	42.797	37.821	**40.280**	**47.214**	33.154
BD > BD + Random Texture > Previous + Random BG	37.857	18.846	40.717	37.183	43.717	48.594	37.667
BD-S > BD-S + Random BG > Previous + Random Texture	**35.740**	17.590	**40.250**	**36.670**	42.810	48.460	33.520
BD-SL > BD-SL + Random Texture > Previous + Random BG	37.640	19.810	40.930	37.210	46.460	50.700	37.710
BD > BD + Random BG > Previous + Random Texture > Previous + Gaussian Blur	38.494	22.564	44.590	39.942	44.486	48.233	50.164

Table 5.2: 2DKPE (in pixels) for the models trained on mixed datasets and finetuned on datasets with incremental parameter variance on real world datasets (a) Freihands Training (b) HO-3D (c) Large-scale Multiview 3D Hand Pose (MHP) (d) Large-scale Multiview 3D Hand Pose Dataset with augmented BG (MHP (Aug)) (e) STB Counting Gestures (f) STB Random Gestures and (g) Freihands Training Dataset with uniform black BG (* Unsurprisingly, the best error metric (31.013px) was achieved without BG randomization).

A glimpse of the results of the trained model achieving the most effective results are shown in the images below in Figure 5.14. For more results, please download the videos from this Folder.

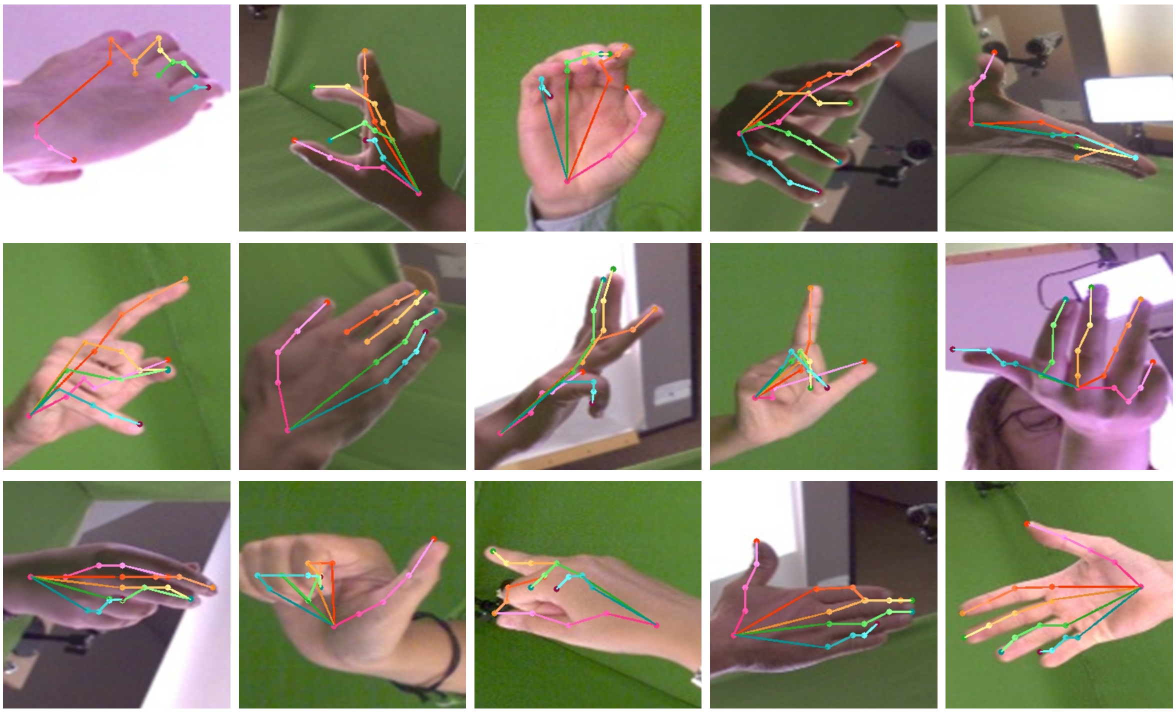


Figure 5.14: Results of the model trained on the finetuning pipeline: Base Dataset with shape variation but no lighting variations (BD-S) > BD-S+Random BG > Previous+Random Hand Texture on the Freihands training dataset.

5.3.3 Ablation Studies

We discuss the effect of individual stages of training and addition of incremental parameter variance with respect to the base dataset without shape or lighting (BD) variations. In case of lighting and shape variance, we had to generate different base datasets, which included these parameter variations. We evaluated the performance of models trained on datasets of increasing parameter variance. To display effect of different parameters, we evaluated these models on the Freihands Dataset. For brevity, in the following ablation study, we show the effect of each stage of finetuning after including the best performing dataset from the previous stage in Figure 5.16.

Our experiments show:

1. The effect on performance as observed in Figure 5.15: Randomization of BG reduces the 2DKPE from 105px to 52px, Randomization of Hand Texture on top of that reduced the 2DKPE further to 35.5px, further addition of S&P Noise reduces this error to 35.1px.
2. In the final stage, where all the variations are introduced through finetuning, performance is in close proximity.
3. The introduction of Gaussian Noise is counterproductive.
4. Figure 5.16, we see the progression that performs well.
5. The positive effect of the shape and lighting variation can be observed from Table 5.2. The 2DKPE of the models trained of base datasets containing shape and/lighting variations performs better than the ones trained on base datasets without shape or lighting variations, both in the full training of the mixed datasets and in the finnetuning pipeline of incremental addition of variance to images in datasets.
6. Additionally, we also notice that most accurate predictions were achieved on the test dataset where the tSNE distribution of our poses overlapped the most with that of the test dataset, i.e, the Freihands Datasets.

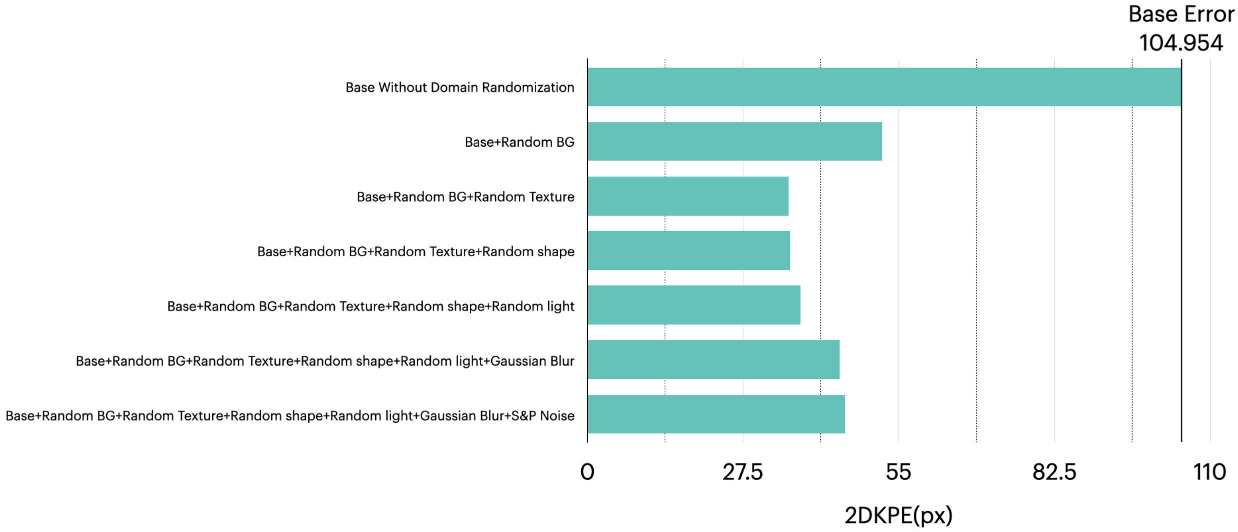
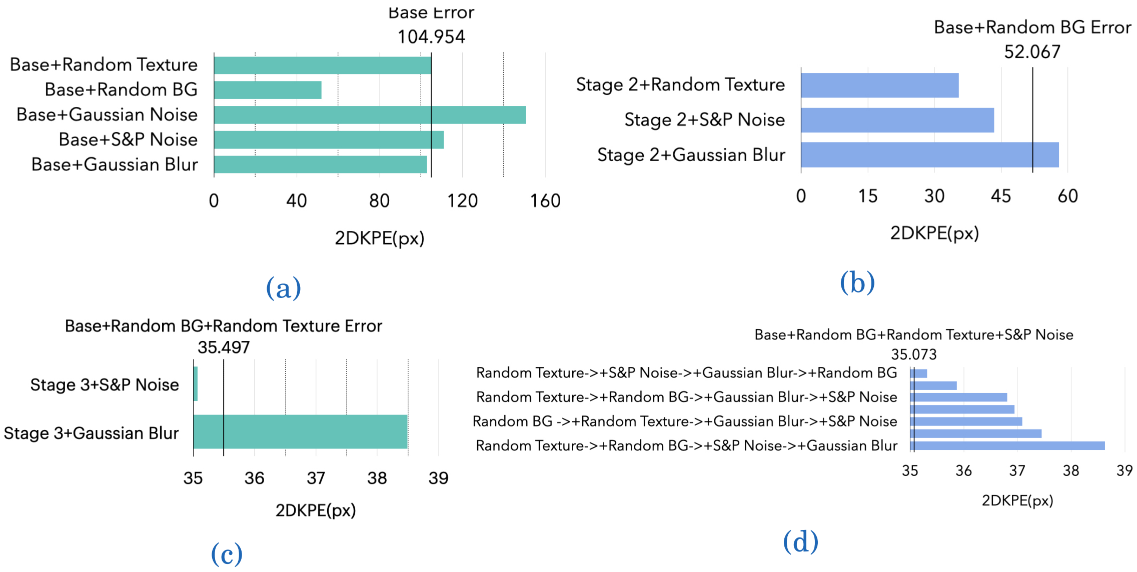


Figure 5.15: Study of effects of randomization of different parameters of the training images by evaluating the 2DKPE on the Freihands Dataset through a finetuning training process.



(a)

(b)

(c)

(d)

Figure 5.16: Ablation studies on the Freihands Dataset with incremental parameter variance through finetuning in (a) Stage 2: on the base dataset without shape or lighting variation, through the addition of Random BG, Random Hand Texture, Salt and Pepper Noise, Gaussian Noise or Gaussian Blur. (b) Stage 3: on the base dataset without shape or lighting variation + Random BG through the addition of Random Texture, Salt and Pepper Noise or Gaussian Blur. (c) Stage 4: on the base dataset without shape or lighting variation + Random BG + Random Texture through the addition of Salt and Pepper Noise or Gaussian Blur. (e) Stage 5: through the addition of all types of variations on the base dataset without shape or lighting variation in different finetuning stage permutations.

5.4 Benchmarks

In Table 5.3 provides a comparison of the performance between models trained using a base dataset without any type of Domain Randomization (except for pose variance) and models trained on datasets with Domain Randomization on the Freihands Dataset. We use the best performing models from Table 5.2. We can also view these differences visually by comparing the image samples in Figure 5.17 with the image samples from Figure 5.14. For video benchmark comparison, please go to this Folder. The metrics, image samples and video comparison of our models clearly demonstrate that Domain Randomization improves Hand Pose Estimation.

Model	Freihands	HO-3D	MHP	MHP (Aug)	STB (C)	STB (R)
Base dataset w/out Domain Randomization	105.480	41.137	62.650	61.148	88.044	86.878
Datasets w/ Domain Randomization (Best Metrics across all experiments)	35.740	16.875	40.250	36.670	40.280	47.214

Table 5.3: Results of training on synthetic dataset without Domain Randomization (base dataset without shape and lighting variation) on the real test datasets for the purpose of benchmarking.

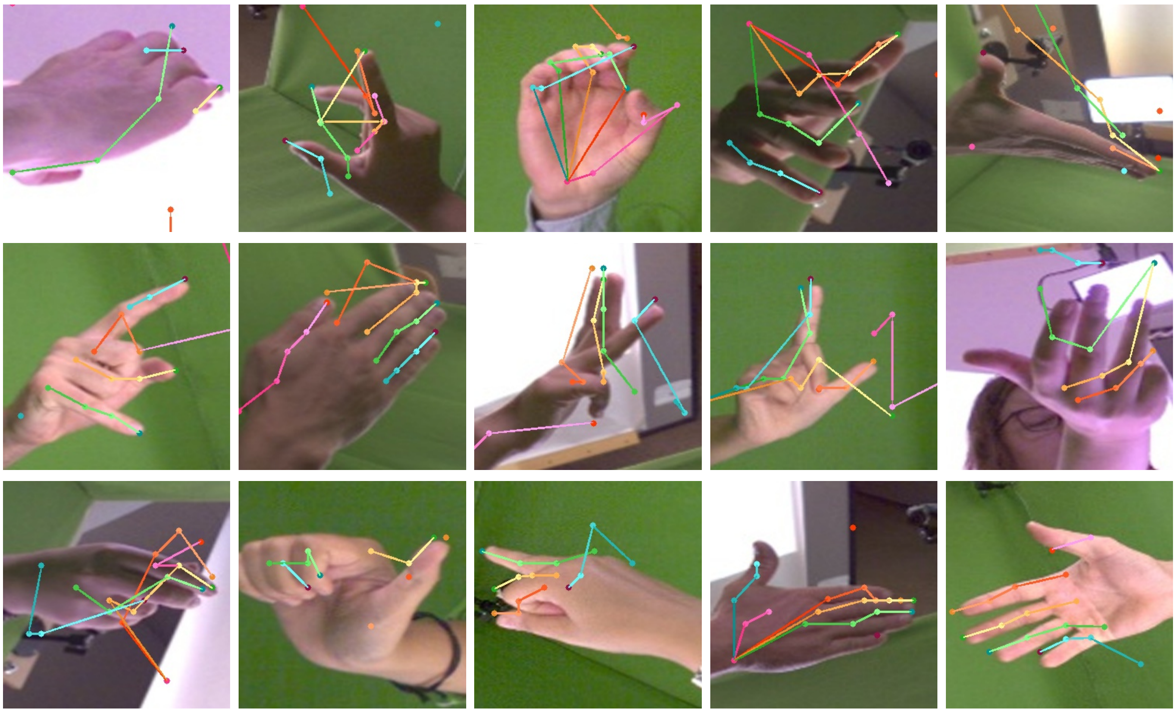


Figure 5.17: Results of training on synthetic dataset without Domain Randomization (base dataset without shape and lighting variation) on the Freihands training dataset.

6 Conclusion and Future Work

Domain Randomization for Hand Pose Estimation successfully bridges the Domain Gap to a large extent when using a training dataset of synthetic images only. It delivers much better results when compared to the performance of a model trained with synthetic images without any Domain Randomization across all the real datasets. While the performance of the models without Domain Randomization (except for pose variance) fails across the datasets for most poses, the models trained with our approach reduces the 2DKPE by 50-68 percent (45-60 pixels), in best case scenarios. Randomization of pose, shape, image background and hand texture achieves large improvements in prediction. There are smaller improvements when Gaussian Blur, S&P Noise or random lighting were introduced in conjunction with the above parameters by shaving off additional 2-3 pixels. The addition of Gaussian Noise is counterproductive. There is scope for improvement when compared to the state-of-the-art methods of Hand Pose Estimation using single RGB images that are trained on real datasets. Future work may include finding more parameters for randomization like randomizing size of hands, introducing other noise models or increasing randomnesss in the parameters that are already mentioned in this work such as the number of PCA parameters of the model to increase pose and shape variance or by increasing the range of online scale augmentation during the training of the model. Intuitively, randomization of lighting could have a more decided advantage on the performance of a model and that could be further explored. In our case, lighting was introduced in the rendering process instead of in post-processing, therefore, we could not isolate it to experiment with the introduction of random lighting in different stages of our finetuning pipeline. In the mixed datasets with both lighting and shape variation, the random lighting was applied to every image, instead of a fraction of the image. From our experiments, the effect of size of datasets remains inconclusive, therefore, one can also attempt to increase the sizes of datasets. We perform all of our experiments on the architecture of OpenPose [7], therefore the performance is limited to the capability of [7]. Hence, another direction of future work could be using different Pose Estimation architectures.

Bibliography

[1] "3dMDhand". In: 2017. URL: http://www.3dmd.com/3dmdhandfoot-system (cit. on p. 11).

[2] Rika Antonova et al. "Reinforcement Learning for Pivoting Task". In: *CoRR* abs/1703.00472 (2017). arXiv: 1703.00472. URL: http://arxiv.org/abs/1703.00472 (cit. on p. 4).

[3] A. Boukhayma, R. de Bem, and P. H. S. Torr. "3D Hand Shape and Pose From Images in the Wild". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10835–10844. DOI: 10.1109/CVPR.2019.01110 (cit. on p. 3).

[4] D. Bowman. "Principles for the design of performance-oriented interaction techniques". In: 2002 (cit. on p. 9).

[5] Richard Boyle et al. "NASA Virtual Glovebox (VGX) - Advanced astronaut training and simulation system for life science experiments aboard the International Space Station". In: Oct. 2001. DOI: 10.2514/6.2001-5105 (cit. on p. 9).

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000) (cit. on pp. 17, 20).

[7] Z. Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1. DOI: 10.1109/TPAMI.2019.2929257 (cit. on pp. 1, 5, 13, 14, 17, 28, 45).

[8] Hyung Jin Chang et al. "Spatio-Temporal Hough Forest for efficient detection–localisation–recognition of fingerwriting in egocentric camera". In: *Computer Vision and Image Understanding* 148 (2016), pp. 87–96. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2016.01.010. URL: http://www.sciencedirect.com/science/article/pii/S1077314216000357 (cit. on p. 9).

[9] J. Y. Chang, G. Moon, and K. M. Lee. "V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5079–5088. DOI: 10.1109/CVPR.2018.00533 (cit. on p. 3).

[10] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on p. 19).

[11] E. Dibra et al. “How to Refine 3D Hand Pose Estimation from Unlabelled Depth Data?” In: *2017 International Conference on 3D Vision (3DV)*. 2017, pp. 135–144. DOI: 10.1109/3DV.2017.00025 (cit. on p. 4).

[12] Bardia Doosti. *Hand Pose Estimation: A Survey*. 2019. arXiv: 1903.01013 [cs.CV] (cit. on pp. 3, 10).

[13] Ali Erol et al. “Vision-Based Hand Pose Estimation: A Review”. In: *Comput. Vis. Image Underst.* 108.1–2 (Oct. 2007), pp. 52–73. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2006.10.012. URL: https://doi.org/10.1016/j.cviu.2006.10.012 (cit. on pp. 3, 8).

[14] Thomas Feix et al. “The GRASP Taxonomy of Human Grasp Types”. In: *IEEE Transactions on Human-Machine Systems* 46 (2016), pp. 66–77 (cit. on p. 11).

[15] L. Ge et al. “3D Hand Shape and Pose Estimation From a Single RGB Image”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10825–10834. DOI: 10.1109/CVPR.2019.01109 (cit. on p. 3).

[16] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. “Large-scale Multiview 3D Hand Pose Dataset”. In: *CoRR* abs/1707.03742 (2017). arXiv: 1707.03742. URL: http://arxiv.org/abs/1707.03742 (cit. on pp. 34, 35).

[17] Y. Hasson et al. “Learning Joint Reconstruction of Hands and Manipulated Objects”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11799–11808. DOI: 10.1109/CVPR.2019.01208 (cit. on p. 4).

[18] Yana Hasson et al. *Learning joint reconstruction of hands and manipulated objects*. 2019. arXiv: 1904.05767 [cs.CV] (cit. on p. 14).

[19] Geoffrey Hinton and Sam Roweis. “Stochastic Neighbor Embedding”. In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS’02. Cambridge, MA, USA: MIT Press, 2002, pp. 857–864 (cit. on p. 12).

[20] D. H. Hubel and T. N. Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of Physiology* 160.1 (1962), pp. 106–154. DOI: 10.1113/jphysiol.1962.sp006837. eprint: https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1962.sp006837. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1962.sp006837 (cit. on p. 12).

[21] Umar Iqbal et al. “Hand Pose Estimation via Latent 2.5D Heatmap Regression”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 125–143. ISBN: 978-3-030-01252-6 (cit. on p. 3).

[22] Y. Jang et al. "3D Finger CAPE: Clicking Action and Position Estimation under Self-Occlusions in Egocentric Viewpoint". In: *IEEE Transactions on Visualization and Computer Graphics* 21.4 (2015), pp. 501–510. DOI: 10.1109/TVCG.2015.2391860 (cit. on p. 10).

[23] Ian Jolliffe. "Principal Component Analysis". In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1094–1096. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_455. URL: https://doi.org/10.1007/978-3-642-04898-2_455 (cit. on p. 12).

[24] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. "Total capture: A 3d deformation model for tracking faces, hands, and bodies". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018 (cit. on p. 13).

[25] M. Kokic, D. Kragic, and J. Bohg. "Learning to Estimate Pose and Shape of Hand-Held Objects from RGB Images". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 3980–3987. DOI: 10.1109/IROS40897.2019.8967961 (cit. on pp. 4, 11).

[26] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539 (cit. on pp. 12, 13).

[27] Sören Lenman, Lars Bretzner, and Björn Thuresson. "Using Marking Menus to Develop Command Sets for Computer Vision Based Hand Gesture Interfaces". In: *Proceedings of the Second Nordic Conference on Human-Computer Interaction*. NordiCHI '02. Aarhus, Denmark: Association for Computing Machinery, 2002, pp. 239–242. ISBN: 1581136161. DOI: 10.1145/572020.572055. URL: https://doi.org/10.1145/572020.572055 (cit. on p. 9).

[28] V. Lepetit, P. Lagger, and P. Fua. "Randomized trees for real-time keypoint recognition". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. 2005, 775–781 vol. 2. DOI: 10.1109/CVPR.2005.288 (cit. on p. 3).

[29] Vincent Lepetit. *Recent Advances in 3D Object and Hand Pose Estimation*. 2020. arXiv: 2006.05927 [cs.CV] (cit. on pp. 3, 10).

[30] J. P. Lewis, Matt Cordner, and Nickson Fong. "Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation". In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172. ISBN: 1581132085. DOI: 10.1145/344779.344862. URL: https://doi.org/10.1145/344779.344862 (cit. on p. 8).

[31] A. Liu et al. "A Survey of Surgical Simulation: Applications, Technology, and Education". In: *Presence* 12.6 (2003), pp. 599–614. DOI: 10.1162/105474603322955905 (cit. on p. 9).

[32] Matthew M. Loper and Michael J. Black. “OpenDR: An Approximate Differentiable Renderer”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 154–169. ISBN: 978-3-319-10584-0 (cit. on p. 18).

[33] Matthew Loper et al. “SMPL: A Skinned Multi-Person Linear Model”. In: *ACM Trans. Graph.* 34.6 (Oct. 2015). ISSN: 0730-0301. DOI: 10.1145/2816795.2818013. URL: https://doi.org/10.1145/2816795.2818013 (cit. on p. 11).

[34] M. Oberweger and V. Lepetit. “DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation”. In: *Proc. of International Conference on Computer Vision Workshops*. 2017 (cit. on pp. 1, 3).

[35] Laurens van der Maaten and Geoffrey Hinton. *Visualizing data using t-SNE*. 2008 (cit. on p. 12).

[36] C. Maggioni and B. Kämmerer. “GestureComputer – History, Design and Applications”. In: *Computer Vision for Human-Machine Interaction*. Ed. by Roberto Cipolla and AlexEditors Pentland. Cambridge University Press, 1998, pp. 23–52. DOI: 10.1017/CBO9780511569937.004 (cit. on p. 3).

[37] Anders Markussen, Mikkel Ronne Jakobsen, and Kasper Hornbaek. “Vulture: A Mid-Air Word-Gesture Keyboard”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Canada: Association for Computing Machinery, 2014, pp. 1073–1082. ISBN: 9781450324731. DOI: 10.1145/2556288.2556964. URL: https://doi.org/10.1145/2556288.2556964 (cit. on p. 9).

[38] I. Mordatch, K. Lowrey, and E. Todorov. “Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 5307–5314. DOI: 10.1109/IROS.2015.7354126 (cit. on p. 4).

[39] F. Mueller et al. “GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 49–59. DOI: 10.1109/CVPR.2018.00013 (cit. on pp. 4, 11).

[40] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. “Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints”. In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV ’11. USA: IEEE Computer Society, 2011, pp. 2088–2095. ISBN: 9781457711015. DOI: 10.1109/ICCV.2011.6126483. URL: https://doi.org/10.1109/ICCV.2011.6126483 (cit. on p. 8).

[41] Xingchao Peng et al. “Learning Deep Object Detectors from 3D Models”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, 2015, pp. 1278–1286. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.151. URL: https://doi.org/10.1109/ICCV.2015.151 (cit. on p. 4).

[42] Zhuwei Qin et al. “How convolutional neural networks see the world — A survey of convolutional neural network visualization methods”. In: *Mathematical Foundations of Computing* 1 (2018), p. 149. ISSN: A0000-0001. DOI: 10.3934/mfc.2018008. URL: http://aimsciences.org//article/id/324b6e02-74f8-4511-ae70-636b3cc0362f (cit. on p. 13).

[43] M. Rad, M. Oberweger, and V. Lepetit. “Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4663–4672. DOI: 10.1109/CVPR.2018.00490 (cit. on pp. 4, 11).

[44] A. Rajeswaran et al. “EPOpt: Learning Robust Neural Network Policies Using Model Ensembles”. In: *ArXiv* abs/1610.01283 (2017) (cit. on p. 4).

[45] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130883. URL: https://doi.org/10.1145/3130800.3130883 (cit. on pp. 1, 5, 8, 11, 17, 18).

[46] Marco Santello, Martha Flanders, and John F. Soechting. “Postural Hand Synergies for Tool Use”. In: *Journal of Neuroscience* 18.23 (1998), pp. 10105–10115. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.18-23-10105.1998. eprint: https://www.jneurosci.org/content/18/23/10105.full.pdf. URL: https://www.jneurosci.org/content/18/23/10105 (cit. on pp. 8, 12).

[47] M. Schröder et al. “Real-time hand tracking using synergistic inverse kinematics”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 5447–5454. DOI: 10.1109/ICRA.2014.6907660 (cit. on p. 8).

[48] J. Shotton et al. “Efficient Human Pose Estimation from Single Depth Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (2013), pp. 2821–2840. DOI: 10.1109/TPAMI.2012.241 (cit. on p. 3).

[49] A. Shrivastava et al. “Learning from Simulated and Unsupervised Images through Adversarial Training”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2242–2251. DOI: 10.1109/CVPR.2017.241 (cit. on p. 4).

[50] Tomas Simon et al. “Hand Keypoint Detection in Single Images Using Multiview Bootstrapping”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 4645–4653. DOI: 10.1109/CVPR.2017.494. URL: https://doi.org/10.1109/CVPR.2017.494 (cit. on p. 9).

[51] D. J. Sturman and D. Zeltzer. “A survey of glove-based input”. In: *IEEE Computer Graphics and Applications* 14.1 (1994), pp. 30–39. DOI: 10.1109/38.250916 (cit. on p. 3).

[52] Baochen Sun and Kate Saenko. "From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains". In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014. DOI: http://dx.doi.org/10.5244/C.28.82 (cit. on p. 4).

[53] D. J. Tan et al. "Fits Like a Glove: Rapid and Reliable Hand Shape Personalization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5610–5619. DOI: 10.1109/CVPR.2016.605 (cit. on p. 8).

[54] J. Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 23–30. DOI: 10.1109/IROS.2017.8202133 (cit. on pp. 4, 5, 11).

[55] Matthew Turk. "Gesture Recognition". In: *Computer Vision: A Reference Guide*. Ed. by Katsushi Ikeuchi. Boston, MA: Springer US, 2014, pp. 346–349. ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_376. URL: https://doi.org/10.1007/978-0-387-31439-6_376 (cit. on p. 9).

[56] Chiraz Walha and Adel M. Alimi. "Human-like Modeling and Generation of Grasping Motion Using Multi-Objective Particle Swarm Optimization Approach". In: *International Journal of Computer Science and Information Security* 14 (2016), pp. 694–710 (cit. on p. 19).

[57] C. Wan et al. "Dense 3D Regression for Hand Pose Estimation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5147–5156. DOI: 10.1109/CVPR.2018.00540 (cit. on p. 3).

[58] C. Wan et al. "Self-Supervised 3D Hand Pose Estimation Through Training by Fitting". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10845–10854. DOI: 10.1109/CVPR.2019.01111 (cit. on p. 4).

[59] S. Wei et al. "Convolutional Pose Machines". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4724–4732. DOI: 10.1109/CVPR.2016.511 (cit. on pp. 13, 14).

[60] D. Xiang, H. Joo, and Y. Sheikh. "Monocular Total Capture: Posing Face, Body, and Hands in the Wild". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10957–10966. DOI: 10.1109/CVPR.2019.01122 (cit. on pp. 13, 28).

[61] F. Xiong et al. "A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation From a Single Depth Image". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 793–802. DOI: 10.1109/ICCV.2019.00088 (cit. on p. 3).

[62] Fang Yin, Xiujuan Chai, and Xilin Chen. "Iterative Reference Driven Metric Learning for Signer Independent Isolated Sign Language Recognition". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 434–450. ISBN: 978-3-319-46478-7 (cit. on p. 9).

[63] Wenhao Yu, C. Karen Liu, and Greg Turk. “Preparing for the Unknown: Learning a Universal Policy with Online System Identification”. In: *CoRR* abs/1702.02453 (2017). arXiv: 1702.02453. URL: http://arxiv.org/abs/1702.02453 (cit. on p. 4).

[64] Jiawei Zhang et al. “3D Hand Pose Tracking and Estimation Using Stereo Matching”. In: *ArXiv* abs/1610.07214 (2016) (cit. on pp. 33, 34).

[65] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. In: *arXiv:1801.09847* (2018) (cit. on p. 27).

[66] J. Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244 (cit. on p. 4).

[67] Christian Zimmermann and Thomas Brox. “Learning to Estimate 3D Hand Pose from Single RGB Images”. In: *IEEE International Conference on Computer Vision (ICCV)*. https://arxiv.org/abs/1705.01389. 2017. URL: https://lmb.informatik.uni-freiburg.de/projects/hand3d/ (cit. on pp. 1, 3, 14).

[68] Christian Zimmermann et al. *FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images*. 2019. arXiv: 1909.04349 [cs.CV] (cit. on pp. 32, 33).