2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506760

# SEMI-SUPERVISED LEARNING OF MONOCULAR 3D HAND POSE ESTIMATION FROM MULTI-VIEW IMAGES

*Markus Müller, Georg Poier, Horst Possegger, Horst Bischof*

Institute of Computer Graphics and Vision, Graz University of Technology

## ABSTRACT

Most modern hand pose estimation methods rely on Convolutional Neural Networks (CNNs), which typically require a large training dataset to perform well. Exploiting unlabeled data provides a way to reduce the required amount of annotated data. We propose to take advantage of a geometry-aware representation of the human hand, which we learn from multi-view images without annotations. The objective for learning this representation is simply based on learning to predict a different view. Our results show that using this objective yields clearly superior pose estimation results compared to directly mapping an input image to the 3D joint locations of the hand if the amount of 3D annotations is limited. We further show the effect of the objective for either case, using the objective for pre-learning as well as to simultaneously learn to predict novel views and to estimate the 3D pose of the hand.

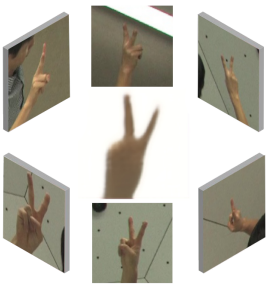
***Index Terms*—** hand pose estimation, convolutional neural networks, novel view synthesis, semi-supervised training

## 1. INTRODUCTION

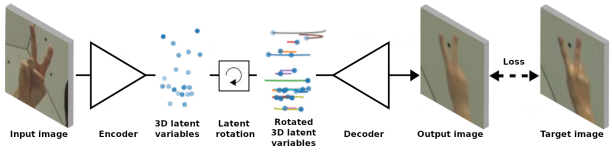
The difficulty of using Convolutional Neural Networks (CNNs) for hand pose estimation is that they require large amounts of labeled training data to work reliably. However, it is a tremendous effort to generate large-scale realistic RGB datasets with accurate 3D annotations.

Therefore, many contemporary works rely on depth images for hand and body pose estimation. A lot of the deep learning-based approaches for 3D hand pose estimation using single depth maps, directly regress the 3D coordinates of the keypoints from the input depth map [1, 2, 3]. Depth-based methods perform very well for 3D hand pose estimation, however, depth cameras are not as commonly available as regular cameras, they only work reliably in indoor environments and have a higher energy consumption.

Neural networks that use only RGB images as input for hand pose estimation can be used across a huge number of devices. However, relying only on RGB images significantly complicates the task. Hence, training a network utilizing RGB images requires even more data compared to training a similar network using depth maps, because of intrinsic ambiguities caused by the visual appearance of the hand under different viewpoints, illumination and occlusions.



(a) Unlabeled multi-view images




(b) Unsupervised learning for our 3D hand pose estimation

**Fig. 1**. Unsupervised learning objective for our 3D hand pose estimation pipeline. **(a)** We use unlabeled multi-view images to learn a geometry-aware representation. **(b)** Our encoder-decoder network is trained in an unsupervised manner. The colored lines in the rotated 3D latent variables indicate the trajectory of the 3D points.

The major bottleneck for many machine learning and computer vision tasks is building an annotated dataset, that is sufficiently large. To reduce the amount of annotated data needed, semi-supervised methods using depth images [4, 5, 6, 7] or synthetically generated data [8, 9] are often used.

We propose a semi-supervised method, that learns to estimate the 3D pose of a monocular hand image, by training a CNN with multi-view RGB images. We build upon the work of Rhodin et al. [10] on 3D human pose estimation which we adapt and optimize to work for hands. We train an encoder-decoder network to learn an unsupervised geometry-aware representation for 3D hand pose estimation using multi-view RGB images for training. We show that using this method, we can achieve similar or better results using significantly less annotated data.

## 2. GEOMETRY-AWARE 3D HAND POSE ESTIMATION

We build upon the approach of Rhodin et al. [10] and use images of the same hand taken from multiple viewpoints to train an encoder-decoder network, which learns a latent representation, that captures the 3D geometry of the hand. We train the encoder-decoder to predict an image seen from one view given an image captured from a different view, without any 2D or 3D pose annotations, as shown in Figure 1. We then use the latent representation to learn a mapping to the 3D pose in a supervised manner. This mapping is much simpler, since the latent representation already captures 3D geometry, and hence requires considerably fewer examples for learning the mapping.

The latent representation $\mathbf{L}$ needs to encode the 3D pose along with shape and appearance, and can be learned without 2D or 3D pose annotations. We use sequences of images acquired from multiple synchronized and calibrated cameras. From these images, we learn separate representations of the hands' 3D pose and geometry $\mathbf{L}^{\text{3D}}$, its appearance $\mathbf{L}^{\text{app}}$, and the background $\mathbf{B}$.

Assuming we are given a set $\mathcal{U} = (\mathbf{I}_t^i, \mathbf{I}_t^j)_{t=1}^{N_u}$ of $N_u$ image pairs without annotations, where *i* and *j* refer to the cameras used to capture the images, and *t* denotes the acquisition time. Let $\mathbf{R}^{i\to j}$ be the rotation matrix from the coordinate system of camera *i* to that of camera *j*. With this basis, we can turn to learning the individual components of $\mathbf{L} = \{\mathbf{L}^{\text{3D}}, \mathbf{L}^{\text{app}}\}$.

To learn the latent representation from an individual image in unsupervised settings, we build upon autoencoders, which are a common choice for such tasks [11, 12, 13]. Let $\mathcal{E}_{\theta_e}$ and $\mathcal{D}_{\theta_d}$ be the encoder and decoder respectively, where $\theta_e$ and $\theta_d$ are the weights controlling their behaviour. The encoder $\mathcal{E}_{\theta_e}$ is used to encode an image $\mathbf{I}$ into a latent representation $\mathbf{L} = \mathcal{E}_{\theta_e}(\mathbf{I})$, which is then decoded into the reconstructed image $\hat{\mathbf{I}} = \mathcal{D}_{\theta_d}(\mathbf{L})$. $\theta_e$ and $\theta_d$ are learned by minimizing the reconstruction error over the training set $\mathcal{U}$.

### 2.1. Encoding multi-view geometry

Our method of using multi-view geometry is influenced by Novel View Synthesis (NVS) methods [11, 12, 14, 15], that rely on training encoder-decoder networks on multiple views of the same object. We let $(\mathbf{I}_t^i, \mathbf{I}_t^j) \in \mathcal{U}$ be two images taken from different views but at the same time $t$. We feed the rotation matrix $\mathbf{R}^{i\to j}$ connecting $\mathbf{I}_t^i$ and $\mathbf{I}_t^j$ as an additional input to the encoder and decoder, and train them to encode $\mathbf{I}_t^i$ and resynthesize $\mathbf{I}_t^j$. Therefore, novel views of the corresponding object can be rendered by manipulating the rotation parameter $\mathbf{R}^{i\to j}$. To enforce an explicit encoding of 3D information within the latent variables, we model the latent representation $\mathbf{L}^{\text{3D}} \in \mathbb{R}^{3\times N}$ as a set of *N* points in 3D space. With this architecture, we can model the view change as a 3D rotation through matrix multiplication of the encoder output by the rotation matrix before it is used as input of the decoder. The resulting autoencoder $\mathcal{A}_{\theta_e,\theta_d}$ can be formally written as

$$\hat{\mathbf{I}}_t^j = \mathcal{A}_{\theta_e,\theta_d}(\mathbf{I}_t^i, \mathbf{R}^{i\to j}) = \mathcal{D}_{\theta_d}(\mathbf{R}^{i\to j}\mathbf{L}_{i,t}^{\text{3D}}) \,, \tag{1}$$

where $\hat{\mathbf{I}}_t^j$ is the reconstructed image from view $j$.

At this point, $\mathbf{L}^{\text{3D}}$ not only encodes the 3D geometry, but also the background and the appearance of the hand. To disentangle this information, we follow related work [10, 14, 15] and create two new vectors $\mathbf{L}^{\text{app}}$ and $\mathbf{B}$ for appearance and background, so that $\mathbf{L}^{\text{3D}}$ only describes geometry and pose.

### 2.2. Network training

We randomly select mini-batches of triplets $(\mathbf{I}_t^i, \mathbf{I}_t^j, \mathbf{I}_{t'}^k)$ in $\mathcal{U}$ with $t \neq t'$ from individual sequences to train $\mathcal{A}$. The first two views are taken at the same time but from different viewpoints and the third one is taken at a different time, but from an arbitrary viewpoint $k$. We need $\mathbf{I}_{t'}^k$ to prevent the network from learning the appearance of the hands. The loss function of the autoencoder is the sum of the pixel-wise loss

$$L_{\theta_e,\theta_d} = |\mathcal{A}(\mathbf{I}_t^i,\, \mathbf{R}^{i\to j},\, \mathbf{L}_{k,t'}^{\text{app}},\, \mathbf{B}_j) - \mathbf{I}_t^j| \,, \tag{2}$$

and a feature loss

$$L_{feat} = |\mathcal{R}_{18}(\mathcal{A}(\mathbf{I}_t^i,\, \mathbf{R}^{i\to j},\, \mathbf{L}_{k,t'}^{\text{app}},\, \mathbf{B}_j)) - \mathcal{R}_{18}(\mathbf{I}_t^j)| \,, \tag{3}$$

which is obtained by first applying a Residual Neural Network (ResNet) [16] $\mathcal{R}_{18}$ with 18 layers trained on ImageNet [17] on the output and target image. This additional term enhances the decodings and improves the pose reconstruction.

$\mathbf{L}_{k,t'} = (\mathbf{L}_{k,t'}^{\text{3D}},\, \mathbf{L}_{k,t'}^{\text{app}})$ is the output of the encoder $\mathcal{E}_{\theta_e}$ applied to the image $\mathbf{I}_{t'}^k$. $\mathbf{B}_j$ is the background in view $j$ and $\mathbf{R}^{i\to j}$ stands for the rotation matrix from view $i$ to view $j$. The encoder $\mathcal{E}$ is applied twice, in order to get $\mathbf{L}_{i,t}^{\text{3D}}$ and $\mathbf{L}_{k,t'}^{\text{app}}$, while ignoring $\mathbf{L}_{i,t}^{\text{app}}$ and $\mathbf{L}_{k,t'}^{\text{3D}}$ for the decoder.

### 2.3. 3D hand pose estimation

Since $\mathbf{L}^{\text{3D}}$ is a $3 \times N$ matrix, it can be understood as a set of $N$ 3D points. To give those points a semantic meaning, we want to derive a pre-defined representation, like a skeleton with $K = 21$ hand joints, encoded as a vector $\mathbf{P} \in \mathbb{R}^{3K}$. To accomplish this, we learn a mapping $\mathcal{F} : \mathbf{L}^{\text{3D}} \to \mathbb{R}^{3K}$. Learning a mapping from $\mathbf{L}^{\text{3D}}$ to the 3D hand joints requires a much smaller amount of annotated data, than what would be needed for learning a mapping directly from the images, as in many other approaches to hand pose estimation.

Let $\mathcal{L} = \{(\mathbf{I}_t, \mathbf{P}_t)\}_{t=1}^{N_s}$ be a small set of $N_s$ labeled examples consisting of images and corresponding ground-truth 3D poses $\mathbf{P}$. Since the latent variable $\mathbf{L}^{\text{3D}}$ already encodes the 3D hand pose and shape, $\mathcal{F}$ is modeled as a Fully Connected

Neural Network (FCNN) with parameters $\theta_f$. It is trained by minimizing the objective function

$$F_{\theta_f} = \lambda_F \frac{1}{N_s} \sum_{t=1}^{N_s} \|\mathcal{F}_{\theta_f}(\mathbf{L}_t^{3D}) - \mathbf{P}_t\|^2 \,, \tag{4}$$

with $(\mathbf{L}_t^{3D}, \cdot) = \mathcal{E}_{\theta_e}(\mathbf{I}_t)$, since the encoder outputs both $\mathbf{L}^{3D}$ and $\mathbf{L}^{app}$.

The encoder-decoder network (trained in an unsupervised manner) introduced in Section 2.2 and the FCNN (trained in a supervised manner) combined form the pre-trained setup, as illustrated in Figure 2a. The unsupervised representation learning already does a lot of the hard work of the challenging task of lifting the image to a 3D representation, that simplifies the final mapping.

In contrast to separately pre-training the latent representation and the mapping to the pose output, our semi-supervised network (see Figure 2b) simultaneously optimizes $\theta_e$ and $\theta_d$ along with $\theta_f$. This setup reduces the training time by almost half. For the majority of our tested scenarios, we have a huge amount of images without 3D annotations and only a small amount of annotated samples. To compensate for the imbalance in our two classes during the joint training procedure, we implemented a random minority oversampling method [18], where we replicate selected samples from the set of annotated images, to achieve a consistent distribution of annotated and unannotated samples in our training batches.

## 3. EXPERIMENTS

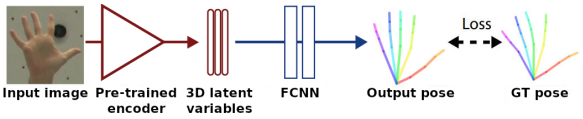
We evaluate our approach on a widely used 3D hand pose estimation benchmark and show that we can achieve more accurate results while using significantly less annotated training data than supervised methods.

### 3.1. Dataset

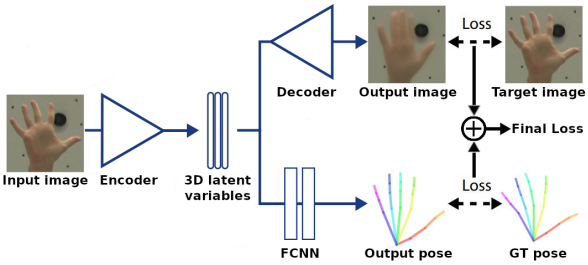
We use the CMU Panoptic[1] dataset [19]. This dataset consists of synchronized camera feeds, calibration, 3D pose reconstruction results, and 3D trajectory streams. The reconstructed 3D hand poses have $K = 21$ keypoints. For our purpose, we use the 31 High Definition (HD) camera feeds (because of the higher resolution), calibration data, and the 3D hand pose reconstructions as ground truth. To train our networks, we employ the sequences *171026_pose3* and *171204_pose2*, and for testing the sequence *171026_pose2*. In total, we have around 24 minutes of video material for training (48 683 frames per camera) featuring 9 different subjects and approximately 9 minutes for testing (16 366 frames per camera) featuring 2 subjects from 31 different HD views.

We first crop the subject of interest to factor out scale and global position. For background estimation, we compute the pixel-wise median over a specific set of frames.

[1] http://domedb.perception.cs.cmu.edu/




(a) Pre-trained network as proposed by Rhodin et al. [10]




(b) Semi-supervised network

**Fig. 2**. Comparison of our network configurations with fixed pre-trained parts (red) and jointly optimized parts for the target task (blue). **(a)** The adopted pipeline from Rhodin et al. [10]. To recover the position of the 3D joints of a hand from a monocular image, we compute the latent representation of the input image and feed it to the FCNN to compute the pose. **(b)** The semi-supervised network trains the encoder-decoder network and the pose network simultaneously.

### 3.2. Evaluation

Since we have ground-truth 3D data for all of our hand patches, we can easily compare different levels of supervision: unsupervised, semi-supervised, or fully supervised. We compare the following configurations:

1) **Supervised**: The network directly maps an input image to the 3D pose, without pre-training the encoder with unlabeled images. The parameters in the encoder and the pose network are optimized simultaneously.

2) **Pre-trained**: This is the hand pose estimation network from Figure 2a using a pre-trained encoder. We pre-train the encoder once with the dataset described in Section 3.1 and another time with random in-plane rotations applied to increase the diversity of the dataset.

3) **Semi-supervised**: The encoder-decoder network and the pose network are trained simultaneously, as depicted in Figure 2b.

Within the complete dataset, we distinguish between the different actors in the video streams and refer to them as S0, S1, S2, S3, S4, S5, S6, S7, S8, where SN specifies all sequences of the $\mathrm{N}^{\mathrm{th}}$ subject. To provide the required supervision to train the FCNN $\mathcal{F}$ depicted in Figure 2, we define different scenarios ranging from as little as 64 annotations to 28 592 if we use all nine subjects. We evaluate pose predic-

| Scenario | #Annotations | N-MPJPE [mm] ↓ | | | |
|---|---|---|---|---|---|
| | | Supervised | Pre-trained | Pre-trained$_{aug}$ | Semi-Supervised |
| All nine training subjects | 28 592 | **16.46** | 21.17 | 21.42 | 18.13 |
| S0+S1+S2+S3+S4 | 18 624 | **18.22** | 21.80 | 21.62 | 19.37 |
| S0+S1+S2 | 13 136 | **20.46** | 22.32 | 21.70 | 21.15 |
| S0+S1 | 9 392 | 24.09 | 23.81 | **22.95** | 25.10 |
| S0 | 6 512 | 26.82 | 25.08 | **24.02** | 26.99 |
| 50% of S0 | 3 248 | 29.22 | 25.36 | **24.60** | 27.96 |
| 10% of S0 | 640 | 34.30 | 27.40 | **26.07** | 30.62 |
| 5% of S0 | 320 | 38.33 | 28.49 | **26.89** | 32.70 |
| 1% of S0 | 64 | 50.97 | 33.08 | **30.40** | 37.19 |

**Table 1**. Comparison of selected network designs in terms of their N-MPJPE over our defined scenarios. For N-MPJPE lower scores are better (denoted by ↓).

tions in terms of the N-MPJPE [20]. In all cases we use S9 and S10 for testing.

We see in Figure 3, that at a certain number of annotated training samples, the accuracy for the pre-trained network does not substantially increase anymore. The N-MPJPE is much lower for smaller numbers of annotated training samples compared to the supervised and semi-supervised networks. If we pre-train the encoder with augmented unlabeled data (pre-trained$_{aug}$), we achieve an even better accuracy. This performance gain diminishes as the amount of annotated data grows. The semi-supervised network outperforms the supervised network, if very little annotations are available and the pre-trained network if many annotated samples are available. As the amount of 3D annotations grows, the supervised approach slightly surpasses our semi-supervised method.

Table 1 compares the N-MPJPE results and shows, that as soon as we use S0, S1 and S2 (13 136 annotated images) for training, the supervised and semi-supervised networks outperform the pre-trained network. An example of NVS predictions including the latent 3D variables, $\mathbf{L}^{3D}$, and pose predictions using all annotations for training of the semi-supervised network can be seen in Figure 4.

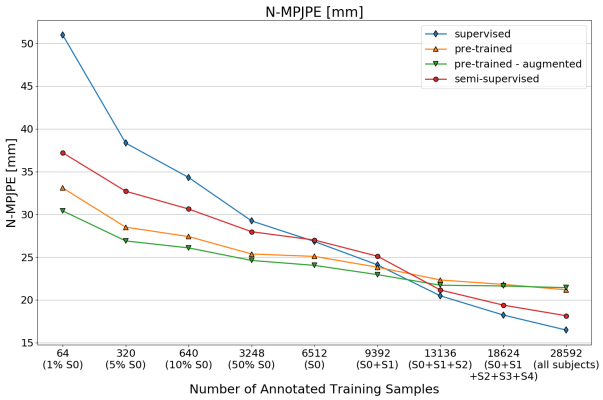



**Fig. 3**. Comparison of the network configurations by their N-MPJPE at different numbers of annotated training data.

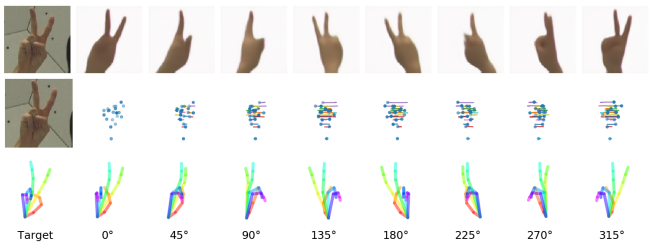



**Fig. 4**. Qualitative results for Novel View Synthesis (NVS), 3D latent variables and pose predictions of the semi-supervised network.

## 4. CONCLUSION

We presented a semi-supervised approach for 3D hand pose estimation, that outperforms fully-supervised methods given a limited amount of labeled data. We trained an encoder-decoder network with only unlabeled multi-view images in an unsupervised manner to learn a geometry-aware latent representation. This geometry-aware hand representation is effective as an intermediate representation for NVS and 3D hand pose estimation. We showed that using a geometry-aware representation in a pre-trained and semi-supervised approach for 3D hand pose estimation performs much better than methods that do not use this intermediate step, when only few annotated data is available. The performance gain even increased, as we used augmented multi-view images to pre-train the encoder-decoder network. When we had more than 13 000 annotations available during training, the fully-supervised and our semi-supervised method outperformed the approach that uses a pre-trained encoder. Further, training our semi-supervised network takes less time than training the pre-trained pose estimation approach, since we do not have to learn the latent representation in an intermediate step.

## 5. REFERENCES


[1] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] S. Baek, K. I. Kim, and T. Kim, "Augmented skeleton space transfer for depth-based hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3d convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] G. Poier, M. Opitz, D. Schinagl, and H. Bischof, "MURAUER: mapping unlabeled real data for label austerity," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[5] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] N. Neverova, C. Wolf, F. Nebout, and G. W. Taylor, "Hand pose estimation through semi-supervised and weakly-supervised learning," *Computer Vision and Image Understanding (CVIU)*, vol. 164, pp. 56–67, 2017.

[7] Y. Chen, Z. Tu, L. Ge, D. Zhang, R. Chen, and J. Yuan, "So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[8] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "GANerated hands for real-time 3d hand tracking from monocular RGB," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single RGB images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation learning for 3d human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[11] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Single-view to multi-view: Reconstructing unseen views with a convolutional network," *Computing Research Repository (CoRR)*, vol. abs/1511.06702, 2015.

[12] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[13] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3d view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] T. S. Cohen and M. Welling, "Transformation properties of learned visual representations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[15] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[18] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[19] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. C. Nabbe, I. A. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social interaction capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 1, pp. 190–204, 2019.

[20] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 7, pp. 1325–1339, 2014.