A Deep Learning Based Paradigm in 3D Human Pose Detection and Estimation in Multi-View Videos

ECE445/ME470 Design Document

Fengkai Chen, Feiyu Zhang, Zhuoting Han, Han Zheng

Group 14

Project Advisor: Gaoang Wang

Course Instructor: Mark Butala

1.Introduction

1.1 Problem and Solution Overview

Human Pose estimation and reconstruction is a widely researched topic in the recent decades. Its main idea is detecting location of people's joints which form a skeleton, and to estimate the posture and movement of human body. Estimating the pose of a human in 3D given an image or a video has recently received significant attention from the scientific community. The main reasons for this trend are the ever-increasing new range of applications (e.g., human-robot interaction, gaming, sports performance analysis) which are driven by current technological advances [1].

Although recent approaches have reported remarkable results in 3D pose estimation from static images, it remains an unsolved problem in continues-time videos. This is because the time-varying overlaps of human bodies in consecutive video frames impose several challenges in detecting the joints from human bodies, which are not fully addressed by existing methods.

The objective of this project is to propose a 3D Pose Estimation paradigm for video setting via leveraging machine learning and optimization technique. In particular, we will first use a Neural Network (NN) to detect the human body (pose) from the surroundings in video clips captured by our multi-view camera system. The detected poses are indicated by a group of boxes (bounding boxes). Then we apply multi-way matching algorithm to cluster the detected 2D poses in the resulting bounding boxes, and finally reconstruct the 3D pose associated to each person.

The multi-way matching algorithm aims at finding 2D poses of the same person in a group of videos clips captured by several cameras (our camera system). For example, there are five people (labeled in 1 to 5) in the room, and we have three cameras shooting from different directions. Then the multi-way matching algorithm matches the 2D pose of person 1 in video from camera 1 to his 2D poses in videos from camera 2 and camera 3. The matched 2D poses of person 1 are categorized by bounding boxes of a specific color (a color corresponds to a labeled person), and 2D poses inside the bounding boxes with same color will be cut-out from the video for 3D pose reconstruction. The 2D to 3D pose reconstruction is done by some well-developed approach such as 3D pictorial structure (3DPS) based model [2].

A schematic of our design is shown as follows.

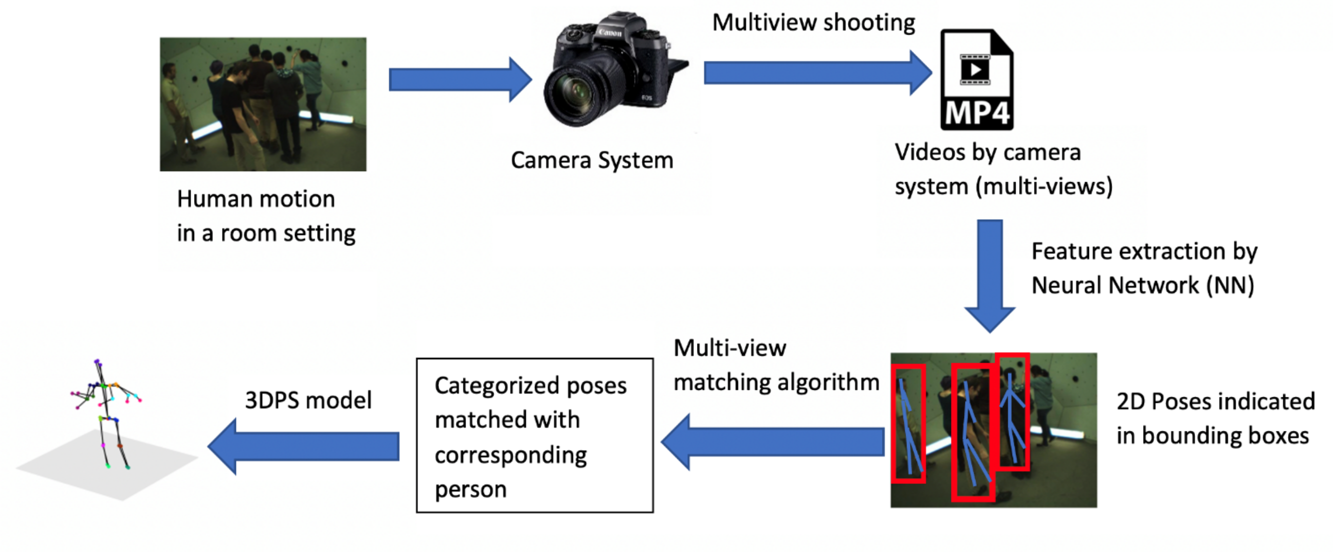

Figure 1.1 Visual Aid: Design schematic

The designing process of this project consists of programming environment configuration (setting up parallel-computing-based programming platform CUDA and machine learning packages such as TensorFlow in Linux system), neural network design (adopting convolutional layer and recurrent layer etc.) and architecture validation (finding the optimal size of network layers, optimal layer concatenations etc.), multi-view matching algorithm implementation, and experiment and demonstration.

As for the metric for evaluating our design, we introduce the Mean Per-Joint Position Error (MPJPE) proposed in [12]. MPJPE is the average Euclidean distance between the location of real-life joints on human bodies and the location of predicted joints on 3D pose model. As mentioned in [7], the MPJPE is as low as 150 (the lower the better) on the dataset Human3.6M (an open-source video data set). Considering we will implement a model that can detect videos instead of photos, the error will be higher. We plan to have similar reconstruction error on the promising datasets like CMU Panoptic [12], Human3.6M and TotalCapture.

1.2 High-level requirements

- Total number of human bodies predicted in 3D pose model should equal to the true number of human bodies in video clips.
- Number of joints in 3D pose of every person should be correct (at least 13, the least number to represent a human 3D pose).
- We expect a MPJPE to be 200(±15) on CMU Panoptic dataset.

2.Design

Block Diagram

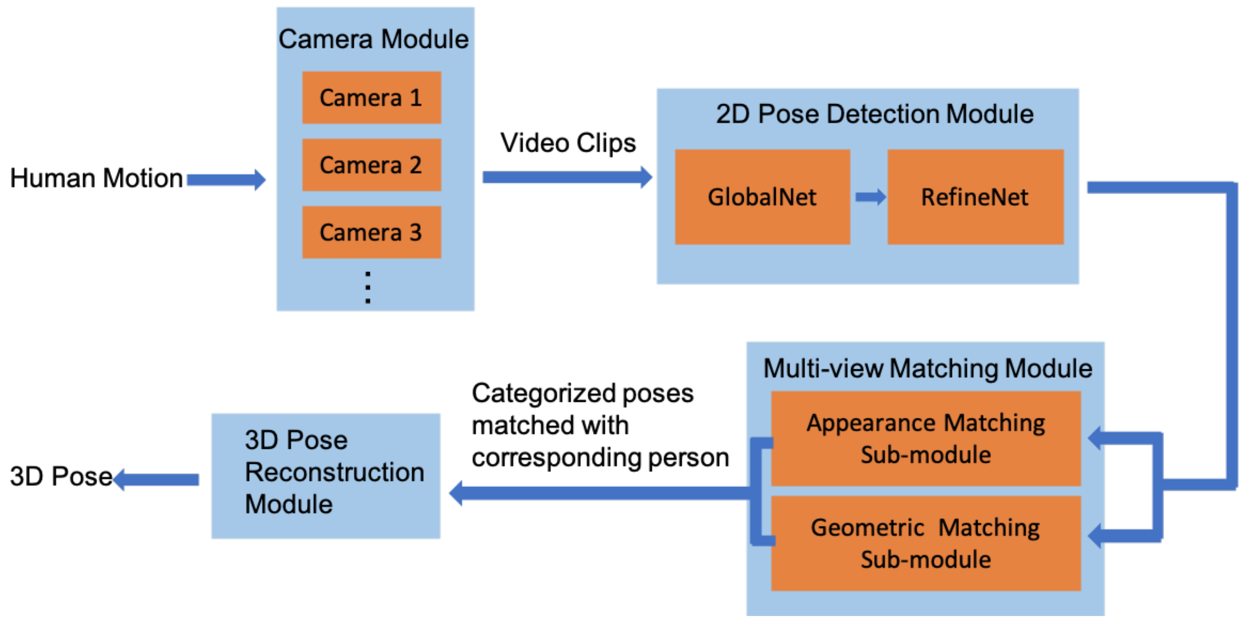

Figure 2.1 Block Diagram

As shown in the block diagram, human motion is captured by camera modules which contains more than three cameras, and stored as video clips. 2D Pose Detection Module allows accurate identification of 2D joints in human bodies from video clips, and Multi-view Matching Module selects the 2D poses of the same person among all 2D poses and groups them. And finally, the 3D Pose Reconstruction Module maps the 2D poses of the same person to his 3D poses. If four main modules mentioned above function properly, all high-level requirements will be achieved.

2.1 Camera Module

Camera module contains several cameras with different positions and projection angles as the following figure shown. The camera system provides us a multi-frame real time video of human movement, which will be used as raw data for subsequent software processing.

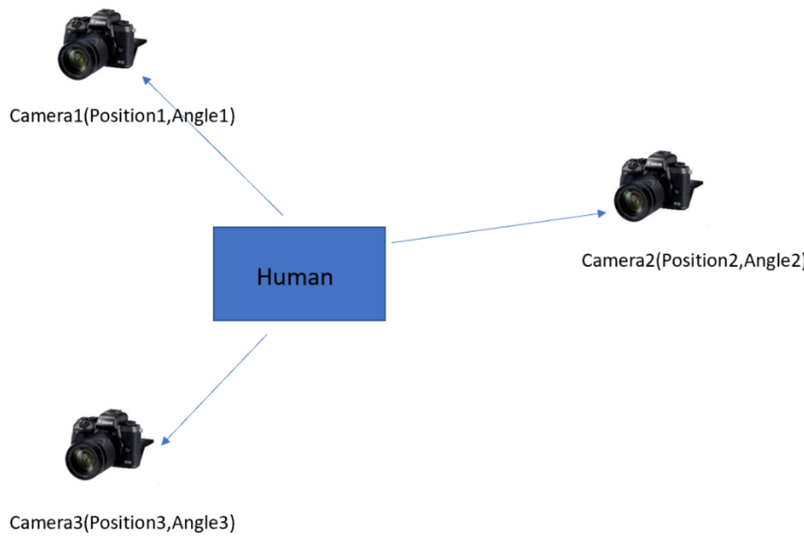

Figure 2.2 Example of Camera System

Requirements	Verification
1.Provide 24 to 30 fps frames video data. 2.Work under 11.5V-15.5V DC voltage supply 3.Maintain normal working status between 0 °C to 40 °C	1A. Measure the real time output data using a laptop, ensuring that the transmitted videos are between 24 and 30 fps. 2A. Connect the camera system to a 11.5V DC power supply, measured by a voltage meter. 2B. Measure the real time output data using a laptop, ensuring that the transmitted videos are between 24 and 30 fps. 2C. Repeat above process while adjusting the power supply until 15.5V DC.

	3A. While verification for Requirement 1 and 2, use a thermometer to ensure that the temperature of working space is between 0 °C and 40 °C.

Table 2.1 RV Table of Camera System

2.2 2D Pose Detection Module

This 2D pose detection module serves to produce 2D locations of people in each view, and every detected human pose will be labeled with box, which is called “bounding box”. For example, in the figure below, a walking man is detected and marked with some joints to represent his pose. Our result will be a bit different from this figure because we use bounding box to label each human pose. Our approach will be using a Convolutional Neural Network (CNN) pretrained on MSCOCO [10] dataset for 2D pose detection in images to finish this task. As shown in the block diagram, our CNN consists of two sub networks: the GlobalNet estimates human limbs approximately and the RefineNet provides more detailed human joints on 2D poses. Both models are proposed in [14].

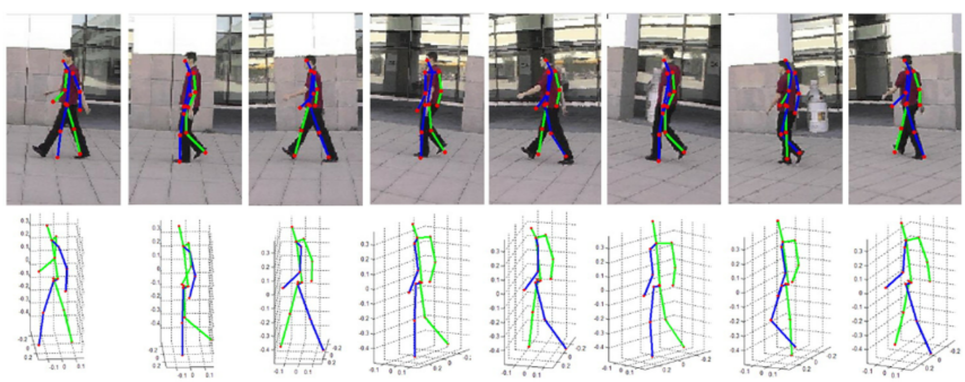

Figure 2.3 Example of 2D Pose Detection Module [9]

Requirements	**Verification**
1. Generate joints and bounding boxes with precision higher than 90% on CMU Panoptic [11] dataset 2. Every human pose must contain at least 13 joints (the least number to represent a human pose using joints)	1A. We use this module for several different test datasets which are selected from CMU Panoptic dataset, and we check every time if the precision is higher than 95%. 2A. After every test of 1A, we will observe each detected human pose to check if the number of joints is higher than 13.

Table 2.2 RV Table of Camera System

2.3 Multi-view Matching Module

Match the detected 2D poses across views, i.e., find in all views the 2D poses belongs to the same person. We will use a discriminative metric to measure the likelihood that two 2D poses belong to the same person and a matching algorithm to establish the correspondences of across multiple views.

2.3.1 Appearance Matching Sub-module

2D pose within bounding boxes is feed to another CNN and the out-put vector of the last layer is taken as the feature of the input pose. We compute the Euclidean distance between two feature vectors and normalize this distance using sigmoid function to range (0,1) as the appearance discriminative score of these two poses. Two poses from the same person should have near-zero appearance discriminative score.

Requirements	**Verification**
1. For images from the same person in multi-view, this sub-module should give a diversity score lower than threshold ε_0 (An empirical value which will be set by some pre-testing) 2. For images from different persons, this sub-module should give a higher diversity score than ε_0.	1A. To begin with, we will do some test on small test datasets to determine threshold ε_0. 1B. After determining threshold ε_0, we will input several images from the same person from our team members to check if the diversity score is lower than threshold ε_0. 2A. After determining threshold ε_0, we will input several images from different persons from our team members to check if the diversity score is higher than ε_0.

Table 2.3 RV Table of Appearance Similarity sub-module

2.3.2 Geometric Matching Sub-module

The multi-view matching function of this sub module is based on the fact that a joint on a body in the first camera view should lie on the epipolar line (the straight line of intersection of the epipolar plane) with the image plane. as sociated with its correspondence in the second camera view, which can be explained in the schematic following:

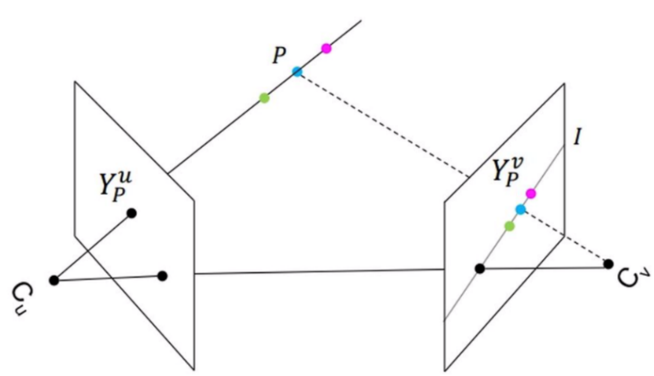

Figure 2.4 Explanation of epipolar line and projection

An image point Y_P^u back-projects to a ray in 3D defined by the camera C_u and Y_P^u. This line is imaged as I in the camera C_v. The 3D point P which projects to Y_P^u must lie on this ray, so the image of P in camera C_v must lie on I.

For two poses, we measured the average point-to-line distance between joints in one pose to the epipolar line associated with these joints in another pose. Then normalize this distance using sigmoid function to range (0,1) as the appearance geometric discriminative score. Two poses from the same person should have near-zero geometric discriminative score. We take the product of geometric discriminative score and appearance discriminative score (mentioned in the previous sub-module) as the discriminative metric to measure the likelihood that two 2D poses belong to the same person. The matching algorithm takes the smallest discriminative score of each pose and identify the another pose which this score corresponds to. Then these two poses should belong to the same person.

Requirements	**Verification**
1. For images from persons in the same position in multi-view, this sub-module should give a diversity score lower than threshold ε_1 (An empirical value which will be set by some pre-testing) 2. For persons from different positions, this sub-module should give a higher diversity score than threshold ε_1	1A. To begin with, we will do some test on small test datasets to determine threshold ε_1. 1B. After determining threshold ε_1, we will input several images from the person of the same position to check if the diversity score is lower than threshold ε_1. 2A. After determining threshold ε, we will input several images from persons from our team of different positions members to check if the diversity score higher than threshold ε_1.

Table 2.4 RV Table of Geometric Discriminative Sub-module

2.4 3D Pose Reconstruction Module

Actually, given the estimated 2D poses of the same person from different views, we can directly reconstruct 3D pose by simple triangulation. However, the error in 2D pose estimation may significantly degrade the reconstruction process. So we are going to use 3D pictorial structure(3DPS) [13] model to reconstruct the 3D pose. 3DPS is kind of similar to search algorithm, which search in the 3D space for the points with highest posterior probability that may occur. The theorem is like following:

Let $S = \{s_i|\ i = 1,2,..N\}$, where $s_i \in R^3$ denotes the predicted 3D position of joint i on the reconstructed 3D pose. If we have 2D poses from total of M camera views, ie. $R = \{r_j|\ j = 1,2,..M\}$. Then we have the posterior distribution of 3D poses can be written as:

$$P(S|R) \propto \prod_{j}^{V} \prod_{i=1}^{N} P\left(r_j \middle| \pi_j(s_i)\right)$$

Where $\pi_j(s_i)$ denotes the 2D projection of s_i the view of camera j. We get $P\left(r_j \middle| \pi_j(s_i)\right)$ by feeding the grouped 2D poses from the Multi-view Matching Module to the Neural Network Model proposed in [14], which determines the 2D spatial distribution of each joint. Then the optimal 3D pose reconstruction S^* is achieved by maximizing $P(S|R)$ via searching among all possible predicted joints placement in 3D space.

Requirements	Verification
1. Given a set of 2D poses from the same person in different views, 3D Pose Reconstruction Module should be able to generate a single 3D pose in 3D space that has a correct joints number (at least 13). 2. Given 2D poses that are matched into different 3D persons, this module should not reconstruct them into a single 3D pose.	1A. We will use several test datasets that includes pre-matched 2D poses of the same person, then we use this module to generate corresponding 3D pose. After processing, we will manually check if the result is a single 3D pose with correct joints number. 2A. We use pre-matched 2D poses from different 3D persons as input, then we check if the output will generate an error output or wrongly reconstructed single 3D pose.

Table 2.5 RV Table of 3D Pose Reconstruction Module

2.4 Tolerance Analysis

Suppose there are k people in the videos, each comprised of n joints. After processing, we can represent i_{th} people's j_{th} joint with three features $\widehat{y_{i,j}} = (\hat{x}, \hat{y}, \hat{z})$, and we have the ground truth label $\mathrm{y_{i,j}} = (\mathrm{x}, \mathrm{y}, \mathrm{z})$. Then we can use Mean Per Joint Position Error (MPJPE) to validate the performance of our program. Per joint position error is the Euclidean distance between ground truth and prediction for a joint. The prediction error is $Err_{pred} = \frac{1}{k}\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}\left\|\widehat{y_{i,j}} - \mathrm{y_{i,j}}\right\|_2$.

Suppose there is only one person in the video, and we establish the coordinate system based on him, setting his pelvis as the origin. Suppose this person is comprised of 5 joints – pelvis, two feet and two hands. So we have $\widehat{y_{1,1}} = (0,0,0), \widehat{y_{1,2}} = (-0.2, 0, -1), \widehat{y_{1,3}} = (0.2, 0, -1), \widehat{y_{1,4}} = (-0.1, 0, 0.5), \widehat{y_{1,5}} = (0.1, 0, 0.5)$. And the estimation of our program is $\mathrm{y_{1,1}} = (0,0,0.1), \mathrm{y_{1,2}} = (-0.2,0,0), \mathrm{y_{1,3}} = (0.2,0,-1), \mathrm{y_{1,4}} = (-0.1,0,0.5), \mathrm{y_{1,5}} = (0.1,0,0.5)$.

$$Err_{pred} = \frac{1}{k}\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}\left\|\widehat{y_{i,j}} - \mathrm{y}_{\mathrm{i,j}}\right\|_{2} = \frac{1}{5}\left(\sqrt{(0-0.1)^2} + \sqrt{(-1-0)^2}\right) = 0.22$$

3. Cost and Schedule

3.1 Cost Analysis:

3.1.1 Labor Cost

Name	Hourly Rate	Hours	Total	Total*2.5
Fengkai Chen	$30	240	$7200	$18000
Han Zheng	$30	240	$7200	$18000
Zhuoting Han	$30	240	$7200	$18000
Feiyu Zhang	$30	240	$7200	$18000
Total				$72000

Table 3.1 Labor Cost

3.1.2 Parts

Description	Quantity	Manufacturer	Vendor	Cost/Unit	Total Cost
Webcam	5	Logitech	Amazon	$27.47	$137.35
1TB SSD	1	Western Digital	Amazon	$89.99	$89.99
Colab Pro	3	/	Google	$9.99 /Month	$29.97
Total					$257.31

Table 3.2 Parts Cost (Our Project mainly based our camera, PC and Colab)

3.1.3 Grand Total

Section	Total
Labor	$72000
Parts	$145.43
Grand Total	$72257.31

Table 3.3 Grand Total Cost

3.2 Schedule:

	Fengkai Chen	Han Zheng	Zhuoting Han	Feiyu Zhang
03/01/21	Study and read related paper	Study and read related papers	Study and read related paper	Study and read related paper
03/08/21	Set up the environment on PC	Study the network structure in different papers	Set up the environment on PC	Study the network structure in different papers
03/15/21	Use the provided model to perform 3D estimation on Image dataset	Designed multiple network optimizations for video 3D estimation	Use the provided model to perform 3D estimation on Image dataset	Designed multiple network optimizations for video 3D estimation
03/22/21	Examine the first demo results,data dugmentation	Optimize the network， apply LSTM to the current network	Examine the first demo results,data dugmentation	Optimize the network， apply LSTM to the current network
03/29/21	Trainning our model on video dataset.(CMU Panoptic)	Help with model trainning, fix any problems during trainning process	Trainning our model on video dataset.(CMU Panoptic)	Help with model trainning, fix any problems during trainning process
04/05/21	Run the model validation& test. Analyze the result analysis.	Set the data collecting environment. Collect our own dataset using webcam	Run the model validation& test. Analyze the result analysis.	Set the data collecting environment. Collect our own dataset using webcam
04/12/21	Using webcam recorded/real-time videos to demo 3D estimation	Refine the algorithm, try to optimize model's performence	Using webcam recorded/real-time videos to demo 3D estimation	Refine the algorithm, try to optimize model's performence
04/19/21	Try real-time videos under	Fix the problems of real-time	Try real-time videos under	Fix the problems of real-time

	complex situation(Small object, overlap object) to test the robustness of our model	video 3D estimation, try to optimize the performance under complex situation.	different situation(Small object, overlap object) to test the robustness of our model	video 3D estimation, try to optimize the performance under complex situation.
04/26/21	Start with Final Report	Final Test on our model & Result Analysis	Start with Final Report	Final Test on our model & Result Analysis
05/03/21	Prepare Final Presentation & Finish Final Report	Prepare Final Presentation & Finish Final Report	Prepare Final Presentation & Finish Final Report	Prepare Final Presentation & Finish Final Report

Table 3.4 Weekly Schedule

4. Ethics & Safety

Our project has several potential safety and ethics concerns. The first concern is network intrusion. Currently we are using campus network to transmit our information and signals. However, every network has a possibility to be attacked, and this rule also applies to our campus network. This is against #7 and #9 of the IEEE Code of Ethics – "the people committing piracy are not properly crediting the work of others, and they could be injuring the copyright holders by sharing content without paying for it." [4] Once the network is controlled, we may lose our control over the whole system, such that our core codes and algorithms may leak. Actually, we do not have a perfect plan for this. Our current solution is that use version control tools, like SVN and git, to store our codes and do not publish it before some sense of agreement is made.

The second concern is the private pictures/video disclosure. The disclosure violates the ACM code of Ethics, #1.6, "Therefore, a computing professional should become conversant in the various definitions and forms of privacy and should understand the rights and responsibilities associated with the collection and use of personal information." [5] Due to the high volume of picture/videos used for network training, saving all data in our personal laptop is not recommended. For convenience in calling data, we plan to store our data on an online server, which may be cyber-attacked and cause data disclosure. To minimize such risk, we suggest shutting down network acceleration software such as Cisco AnyConnect Mobility Client and Express VPN when testing online algorithms.

With the following concerns are fully considered, we still want to make sure that the model will treat everyone equally. If we use a biased training dataset, like some dataset mostly containing videos/pictures of white people, the model may have worse effects on black, Asian and Hispanic people. If we use a training dataset that mostly involves men moving and acting, this model may have worse effects on women. All these violate the #8 of the IEEE Code of Ethics, "to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression" [4]. To avoid such things, we will carefully choose our dataset, including the percentage of different races, genders, ages and other tags that may divide people into different groups, to ensure an unbiased development process.

References

[1] Sarafianos, Nikolaos & Boteanu, Bogdan & Ionescu, Bogdan & Kakadiaris, Ioannis. 3D Human Pose Estimation: A Review of the Literature and Analysis of Covariates, in *Computer Vision and Image Understanding*, 2016.

[2] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, 2014.

[3] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *CVPR*, 2019.

[4] CVSSP Research. Multi-Person 3D Pose Estimation and Tracking in Sports. Apr.17, 2019. [Online Video]. Available: https://www.youtube.com/watch?v=jLEv14GAcbs

[5] ieee.org, IEEE Code of Ethics, 2020. [Online].

Available: http://www.ieee.org/about/corporate/governance/p7-8.html[Accessed: 28- Feb- 2021].7

[6] ethics.acm.org, ACM Code of Ethics, 2020. [Online]. Available: https://ethics.acm.org. [Accessed: 28- Feb- 2021].7

[7] openaccess.org, 3D human pose estimation in video with temporal convolutions and semi-supervised training, 2020. [Online].

[8] Umar.Iqbal, Pavlo.Molchanov, Jan.Kautz, NVIDIA Research. Weakly-Supervised 3D Human Pose Learning via Multi-view Images in the Wild. In *CVPR*, 2020

[9] Fast Human Pose Detection Using Randomized Hierarchical Cascades of Rejectors - Scientific Figure on ResearchGate.

[10] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[11] Kocabas M, Karagoz S, Akbas E. Self-Supervised Learning of 3D Human Pose using Multi-view Geometry[J]. In *CVPR*, 2019.

[12] Joo, Hanbyul and Simon, et al. Panoptic Studio: A Massively Multiview System for Social Interaction Capture, IEEE Transactions on Pattern Analysis and Machine Intelligence,2017

[13] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele,N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, 2014. 1, 2, 6, 7

[14] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 3, 6, 7