Detection and Prediction of Human Gestures by Probabilistic Modelling

Erkennung und Vorhersage menschlicher Gesten durch probabilistische Modellierung
Master thesis by Lanmiao Liu
Date of submission: May 25, 2021

1. Review: M.Sc. Julen Urain De Jesus
2. Review: Prof. Dr. Jan Peters
3. Review: Jun.Prof. Dr. Philipp Beckerle
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Lanmiao Liu, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Lanmiao Liu

Darmstadt, 25. Mai 2021

Lanmiao Liu

For my grandfather Kejing Liu

Abstract

Supervised learning is among the most successful approaches to classify on gesture of human skeleton. While impressive results were achieved, supervised learning model are limited in new class inference without label as it demands re-training for new class and resulting in more computational. Density estimation is reasonable approach to tackle these problems. A probabilistic view on unsupervised learning improves the inference on new class without re-training on datasets through probability distributions. In this thesis, we discuss the GestureFlow model to classify on gesture of human skeleton inspired by normalizing flow. We evaluate different approaches such as Fully Connected Model, Support Vector Machine and GestureFlow model on our datasets. We further develop a new metric for inference the label on new class. Our results suggest, GestureFlow model is less computational and more powerful expressiveness of probability distributions compared to baseline model.

Zusammenfassung

Überwachtes Lernen gehört zu den erfolgreichsten Ansätzen zur Klassifizierung nach Gesten des menschlichen Skeletts. Während beeindruckende Ergebnisse erzielt wurden, wurde das überwachte Lernmodell in einer neuen Klasseninferenz ohne Label begrenzt, da es eine Umschulung für eine neue Klasse erfordert und zu mehr Rechenaufwand führt. Die Dichteschätzung ist ein vernünftiger Ansatz, um diese Probleme anzugehen. Eine probabilistische Sicht auf unbeaufsichtigtes Lernen verbessert den Rückschluss auf eine neue Klasse, ohne die Datensätze durch Wahrscheinlichkeitsverteilungen erneut zu trainieren. In dieser These diskutieren wir das GestureFlow-Modell, um die Geste des menschlichen Skeletts zu klassifizieren, die durch die Normalisierung des Flusses inspiriert ist. Wir bewerten verschiedene Ansätze wie das vollständig verbundene Modell, die Support Vector Machine und das GestureFlow-Modell in unseren Datensätzen. Wir entwickeln eine neue Metrik weiter, um auf das Etikett für eine neue Klasse schließen zu können. Unsere Ergebnisse legen nahe, dass GestureFlowmodel im Vergleich zum Basismodell weniger rechnerisch und aussagekräftiger für Wahrscheinlichkeitsverteilungen ist.

Contents

1. Introduction

With the development of robotic technology, the impact of robots on our daily lives has become critical. The subsequent interactions between humans and robots will become more and more frequent. The main goal of our work is to model a probability distribution given its sample and classify on human skeleton gesture.

Among the most successful approaches to classify on human skeleton gesture is supervised learning model[1, 2]. For labeled data, it can achieve good classification and has significant classification capabilities. However, for new data sets, a huge amount of computation is required, and it is difficult to infer the new dataset to infer the label.

In unsupervised learning, its importance stems from the relative abundance of unlabeled data compared to labeled data and their applications such as density estimation, outlier detection, prior construction. We follow the normalizing flow[3] to tackle the problem of re-training model for new class. This work further evaluates metrics to measure the classified capabilities of an algorithm.

The struture of thesis shows as follows. In Chapter 2, fundamental concepts required for the rest of this thesis are reviewed briefly. The chapter introduce the the structure of Fully Connected Model, [1] and Fully Connected Model [2] ,and their derivation of classification. Further, we discuss normalizing flow[3] and its powerful expressiveness of probability distributions. Moreover, we give an in-depth introduction to the coupling layer which inspire our GestureFlow model. we describe the structure of neural network of Openpose and its application. Based on Openpose we recording our dataseta and applyit in real-time human skeleton tracking In chapter 3, the related works in the state of the art are presented. In chapter 4, we evaluate GestureFlow model with a comparison to baseline model. In addition, we also considered the classification results learned from the Gaussian distribution and the model learned by normalizing flow. We apply gestureflow model in real-time human skeleton pose classification in 2- and 3-dimensional coordinate. In Chapter 5, We summarized the classification achieved results by our proposed method compared to the baseline model. At the end, in Chapter 6, an outlook for future work is provided to this thesis.

2. Foundations

This chapter introduces the fundamentals in pursuit of algorithms and models. This section consists of two parts. At the beginning of this section, the terms of supervised learning of baseline model get recapped. Furthermore, the different types of generative models are described. Besides, Openpose is introduced for human skeleton detection.

In Section 2.1.1, the basics for the Fully Connected Model(FCM) are provided. We present the basic parameters and show how to apply FCM for learning and classification on human gesture data. In Section 2.1.2, the description of the Support Vector Machine is introduced, which is a supervised model for classification in human gesture data. In Section 2.2.1, Gaussian Naive Bayes Classification is presented. Section 2.2.2 highlights Normalizing Flows, which are unsupervised techniques known for capturing underlying relations in human gesture data.

2.1. Supervised learning of baseline model

2.1.1. Fully Connected Model

Fully connected model is probabilistic models that map the learned distributed feature representation to the sample label space. FCM find their application such as in a wide range of fields such as image classification, voice recognition.

Structure of Fully Connected Model

Fully connected neural network is composed of a range of fully connected layers where each output dimension relies on each input dimension. Figuratively, a fully connected layer is represented based on the architecture from the [4] as follows in Figure.2.1. A neural network is premeditated with i hidden layers, where $i \in\{1,2\}$ index the hidden layers of the network. In the neural network layer, the combined formula of a linear transformation is given in 2.1 as follows:

$$z_i = y_{i-1}w_i + b_i \tag{2.1}$$

where z_i denote the vector of outputs into layer i, y_{i-1} denote the vector of inputs from layer i ($y_0 = x$ is the input), b_i is the biases and w_i weights from layer i. Then the sparse outputs are used as input to the next layer where is applied at each layer. According to this structure of FCM, dropout is applied to a neural network is equivalent to sampling a sparse network from it. Dropout is a technique to solve these two problems. It avoids overfitting and states a way to effectively approximate wide range neural network architectures in the form of combined exponentials.

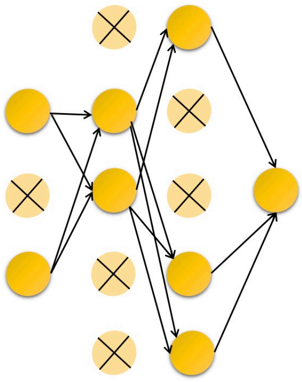


Figure 2.1.: A sparse network created by applying dropout to the two layer network. Crossed units have been discarded.

Moreover, [4] introduce non-linear operation in order to reduce gradient explosion at each layer which is given in 2.2 as follows:

$$y_i = f(z_i) \tag{2.2}$$

where f is a non-linear activation function ReLU. Using the above defined inputs and outputs, the softmax function is considered in the output-layer to normalize the inputs, which is given in 2.3 as follows:

$$\tilde{y}_i = \exp z_i / \sum_i \exp z_i \tag{2.3}$$

we can get a probability distribution of the label of the output. Finally, a cross entropy loss function is evaluated over that dataset in the multi-classification problem. It can be expressed in 2.4 as follows:

$$\mathbf{H}(\mathbf{y}, \tilde{\mathbf{y}}) = -\sum_{z} p(\mathbf{z}) log(q(\mathbf{z})) \tag{2.4}$$

where $p(\mathbf{z})$ is probability of true label of the data, $q(\mathbf{z})$ is probability of predicted label of the data.

2.1.2. Support Vector Machine

This section summarizes the Support Vector Machine used throughout which is a non-parametric supervised learning model. The kernel trick provides the way that maps inputs to high-dimensional feature spaces in non-linear classification. A set of hyper-planes in a high dimensional space is constructed which obtains a good separation. While the distance among nearest training data point of any class is the largest, with the greater the margin, the smaller the generalization error of the classifier. Figure.2.2 shows the decision function of the two samples for a linearly separable issue on the margin boundaries.

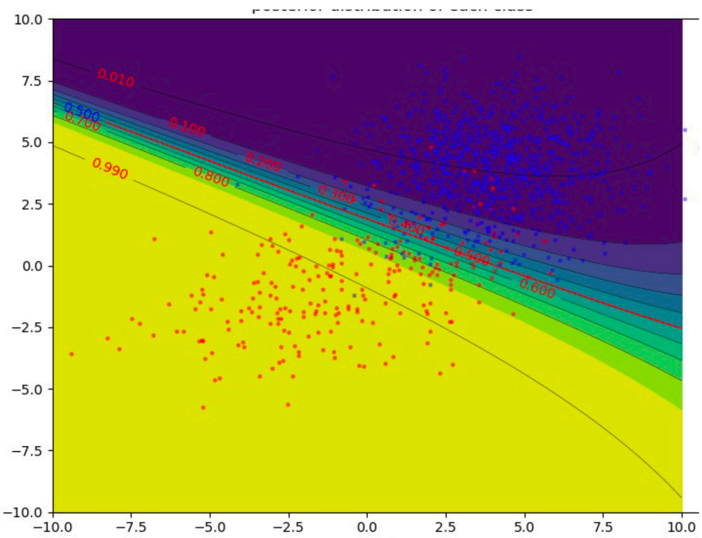


Figure 2.2.: A linearly separable problem in support vector machine.

[2] motivates support vector machines to build a multi-class classifier, in which the k^{th} model $y_k(\mathbf{x})$ is trained using the data from class C_k. Given training vectors , $n = 1, \ldots, t$ in multi-class classes and a vector which means the given prediction is correct for most samples. It solves the following primal problem shown in 2.5 as follows:

$$\begin{aligned} \min_{\mathbf{w}^{ij},\mathbf{b}^{ij},\xi^{ij}} \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C\sum_n \xi_n^{ij} \\ s.t.(\mathbf{w}^{ij})^T\phi(\mathbf{x_n}) + \mathbf{b}^{ij} \geq 1 - \xi_n^{ij}, if\, y_n = i \\ (\mathbf{w}^{ij})^T\phi(\mathbf{x_n}) + \mathbf{b}^{ij} \leq -1 + \xi_n^{ij}, if\, y_n = j \\ \xi_n^{ij} \geq 0, n = 1, ..., t \end{aligned} \tag{2.5}$$

where i and j represent the class of SVM; n the represents the index of the sample in the of the i class and the j class; ϕ represents from input space to feature space of non-linear mapping, ξ is the distance from their correct margin boundary. To maximize the margin $(\mathbf{w}^{ij})^T\mathbf{w}^{ij}$ is minimized under the constraints which obtains optimal hyperplane. Once the optimization problem is solved, the decision function in the feature space will accordingly be of the form:

$$\mathbf{y_{new}^{ij}} = sign[\mathbf{w}^{ij})^T\phi(\mathbf{x_{new}}) + \mathbf{b}^{ij}] \tag{2.6}$$

2.2. Unsupervised learning of generative model

2.2.1. Gaussian Naive Bayes Classification

Naive Bayes Classification

Naive Bayes classification aims at applying Bayes' theorem, which is given the value of class variables. With the “naive” assumption of conditional independence [5] propose a novel explanation on the superb classification performance of naive Bayes. Given $\mathbf{X}$ is represented by a tuple of attribute values $(x_1, x_2, , , x_n)$, where x_i is samples of class variable. $\mathbf{Y}$ is represented by the a tuple of classification variable $(y_1, y_2, , , y_m)$, where y_j

is the label of class variable. From the probability perspective, according to Bayes Rule, the probability of an sample $\mathbf{X} = (x_1, x_2, , , x_n)$ being class y_j is:

$$p(y_j|x_1, ...x_n) = \frac{p(y_j)p(x_1, ...x_n|y_j)}{p(x_1, ...x_n)} \tag{2.7}$$

Naive conditional independence assumption is applied in:

$$p(x_i|y_j, x_1, ...x_{i-1}, ..., x_n) = p(x_i|y_j) \tag{2.8}$$

so that the relationship of given the sample of the class variable is simplified to:

$$p(y_j|x_1, ...x_n) = \frac{p(y_j)\prod_{i=1}^{n} p(x_i|y_j)}{p(x_1, ...x_n)} \tag{2.9}$$

When $p(x_1, ...x_n)$ is constant as given input, the following classification rule is shown:

$$p(y_j|x_1, ...x_n) \propto p(y_j)\prod_{i=1}^{n} p(x_i|y_j) \tag{2.10}$$

$$\hat{\mathbf{y}} = \underset{y_j}{argmax}\, p(y_j)\prod_{i=1}^{n} p(x_i|y_j) \tag{2.11}$$

Maximum A Posteriori (MAP) is applied to estimate $p(y_j)$ and $p(x_i|y_j)$ in the solving the optimization problem, which is deeply related to maximum likelihood. The various of naive Bayes classifiers depend on the assumptions of the distribution of $p(x_i|y_j)$. Although the assumptions of naive Bayes classifiers is oversimplified, it performs well in many practical situations such as document classification and spam filtering. Due to decouple of the class conditional feature distribution helps to alleviate the problems caused by the curse of dimensionality.

Gaussian Naive Bayes

A key component of Gaussian Naive Bayes is the assumptions of the distribution of $p(x_i|y_j)$. The likelihood of the samples is assumed to be Gaussian:

$$p(x_i|y_j) \sim G(\mu, \sigma) \tag{2.12}$$

Maximum likelihood is utilized to estimate the parameters μ and σ.

2.2.2. Normalizing Flows

Normalizing flows[6, 7, 3] represent a family of methods that derives the flexible learnable probability distributions, which which allow us to surpass the limitations of simple parametric forms. The idea behind Normalizing Flows[7] is that operate by pushing a simple density through a serie of transformations to produce a richer, potentially more multi-modal distribution—like a fluid flowing through a set of tubes. The density of the sample can be evaluated by converting the sample back to the original simple distribution, and then calculating the product of the density of the inverse-transformed sample under this distribution and the relative volume change caused by the inverse-transformed sequence[3]. The expressive power of flow-based models and operation of flows in our method are introduced later.

Basics and Property

Considering $\mathbf{x}$ as a D-dimensional sample from a random variable with a known and tractable probability density function $p(\mathbf{x})$. The idea behinds flow-based modeling is to express $\mathbf{x}$ as a transformation $\mathbf{T}$ of $\mathbf{u}$ which is a random variable in the latent space with tractable distribution $p(\mathbf{u})$:

$$\mathbf{x} = \mathbf{T}(\mathbf{u}) \tag{2.13}$$

where u is sampled from $p(\mathbf{u})$:

$$\mathbf{u} \sim p(\mathbf{u}) \tag{2.14}$$

When the transformation $\mathbf{T}$ is invertible and both $\mathbf{T}$ and $\mathbf{T}^1$ are differentiable, transformation $\mathbf{T}$ is defined as diffeomorphisms. In additional, $\mathbf{u}$ is required D-dimensional as well as $\mathbf{x}$, the probability density function of of $\mathbf{x}$ is computed:

$$p(\mathbf{x}) = p(\mathbf{u})|det\mathbf{J_T}(\mathbf{u})|^{-1} \tag{2.15}$$

The partial derivatives of Jacobian matrix $\mathbf{J_T}(\mathbf{u}) \in \mathbb{R}^{D\times D}$ is given in [6, 7, 3]with respect to $\mathbf{u}$:

$$\begin{bmatrix} \frac{\partial T_1}{\partial u_1} & \cdots & \frac{\partial T_1}{\partial u_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_D}{\partial u_1} & \cdots & \frac{\partial T_D}{\partial u_D} \end{bmatrix} \tag{2.16}$$

Identically, the probability density function of of $\mathbf{x}$ is obtained with terms of the Jacobian of $\mathbf{T^{-1}}$:

$$p(\mathbf{x}) = p(\mathbf{T}^{-1}(\mathbf{x}))|det\mathbf{J}_{\mathbf{T}^{-1}}(\mathbf{x})| \tag{2.17}$$

where the transformation of variable of $\mathbf{u}$ is expressed:

$$\mathbf{u} = \mathbf{T}^{-1}(\mathbf{x}) \tag{2.18}$$

Transformation $\mathbf{T}$ is considered as warping the space $\mathbb{R}^D$ in order to mold the density $p(\mathbf{u})$ into $p(\mathbf{x})$[7]. An important property of diffeomorphism transformation is that they are composable. Given $T_1, ..., T_k$ be a set of invertible and dierentiable transformations, then their composition is defined as:

$$\mathbf{T} = T_1 \circ T_2 \circ \cdots \circ T_k \tag{2.19}$$

Its inverse of composition are derived as:

$$\mathbf{T}^{-1} = T_k^{-1} \circ T_{k-1}^{-1} \circ \cdots \circ T_1^{-1} \tag{2.20}$$

Moreover, the determinant of the Jacobian is given by:

$$detJ_{T_k \circ T_{k-1} \circ \cdots \circ T_1}(\mathbf{u}) = detJ_{T_1}(\mathbf{u}) \prod_{k=2}^{K} detJ_{T_k}(T_{k-1}(\mathbf{u}) \circ T_{k-2}(\mathbf{u}) \circ \cdots \circ T_1(\mathbf{u})) \tag{2.21}$$

Complex transformations to be represented by the composition of transformations which consists of a set of simpler transformations[3]. In practice, the determinant of the Jacobian of every simpler transformation can be computed easily.

Sampling from the model and evaluating the model's density are the important operations for a flow-based model[7]. when we sampling from the model, prior distribution $p(\mathbf{u})$ is sampled for computing of the forward transformation $\mathbf{T}$. we evaluate the model's density, inverse transformation $\mathbf{T}^{-1}$ and its Jacobian determinant must be obtained. In practice parameterizing flows can be implemented by deep neural networks.

Coupling Layer

The coupling layer[8, 9] in auto-regressive networks[10, 11] is defined more expressive transformations.To obtain the tractability of Jacobian determinant and simple computation in parameter learning, the architecture of the coupling layer is decisive to achieve the a

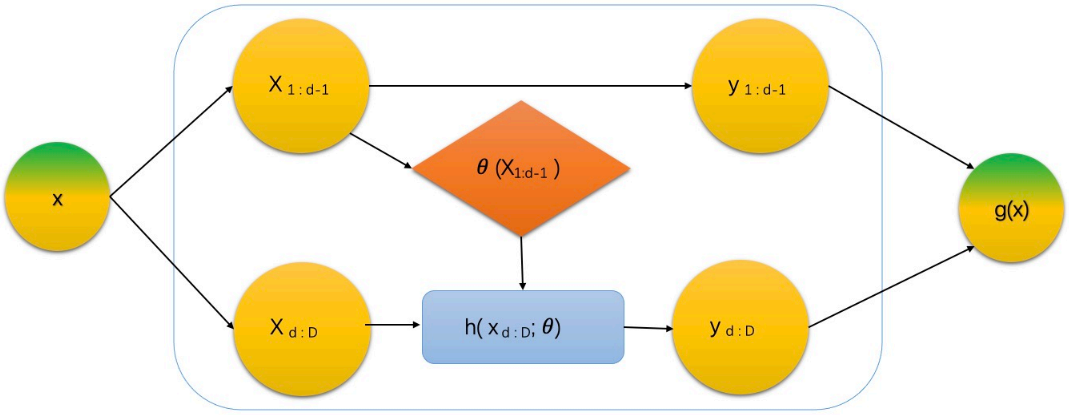


Figure 2.3.: The transformation of coupling layer.

family of bijections. Considering affine transformations, [12] and [13] provide formulas for the inverse and determinant when using diagonal matrices. Inverting triangular matrices at test time is reasonable in terms of computation[8]. With a triangular weight matrix and a bijective activation function it greatly limits the architecture which is designed. Choosing of depth and non-linearity is pessimistic. The triangular Jacobian function is premeditated to ensure the computation of diagonal elements of the Jacobian easily.

Suppose $\mathbf{x} \in \mathbb{R}^D$ be divided into two parts such as $\mathbf{x}_{1:d-1} \in \mathbb{R}^d$ and $\mathbf{x}_{d:D} \in \mathbb{R}^{D-d}$ and a bijection $h(\cdot;\theta) : \mathbb{R}^{D-d} \to \mathbb{R}^{D-d}$, parameterized by θ. The transformation based on coupling layer is defined:

$$g(\mathbf{x}) = \begin{cases} \mathbf{y}_{1:d-1} = & \mathbf{x}_{1:d-1} \\ \mathbf{y}_{d:D} = & h(\mathbf{x}_{d:D};\theta) \end{cases} \tag{2.22}$$

where the resulting function g is called a coupling flow and the parameters θ are defined by any arbitrary function [8]. A coupling flow is invertible when h is invertible:

$$g^{-1}(\mathbf{x}) = \begin{cases} \mathbf{x}_{1:d-1} = & \mathbf{y}_{1:d-1} \\ \mathbf{x}_{d:D} = & h^{-1}(\mathbf{y}_{d:D};\theta) \end{cases} \tag{2.23}$$

The coupling architecture is shown in 2.3: Block triangular Jacobian matrix is defined a:

$$\mathbf{J}_g(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{\partial \mathbf{y}_{d:D}}{\partial \mathbf{x}_{1:d-1}} & \frac{\partial \mathbf{y}_{d:D}}{\partial \mathbf{x}_{d:D}} \end{bmatrix} \tag{2.24}$$

where Jacobian for coupling transformations is a lower and which $\mathbf{I}_d \in \mathbb{R}^{D\times D}$ is the identity matrix. Therefore, Log determinant can be obtained by:

$$log|det\mathbf{J}_g(\mathbf{x})| = log|\prod_{i=d+1}^{D}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_i}| = \sum_{i=d+1}^{D}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_i} \tag{2.25}$$

GestureFlow Model

Our method based on the coupling layer. The pipeline of GestureFlow model is shown in Fig. 2.4. The pipeline of GestureFlow model consists of training block and inference block. In training loop block model libraries is generated from recording data, which derives gesture label and threshold to classify the action of human. In inference loop block model new gestures label is generated from the new human action.

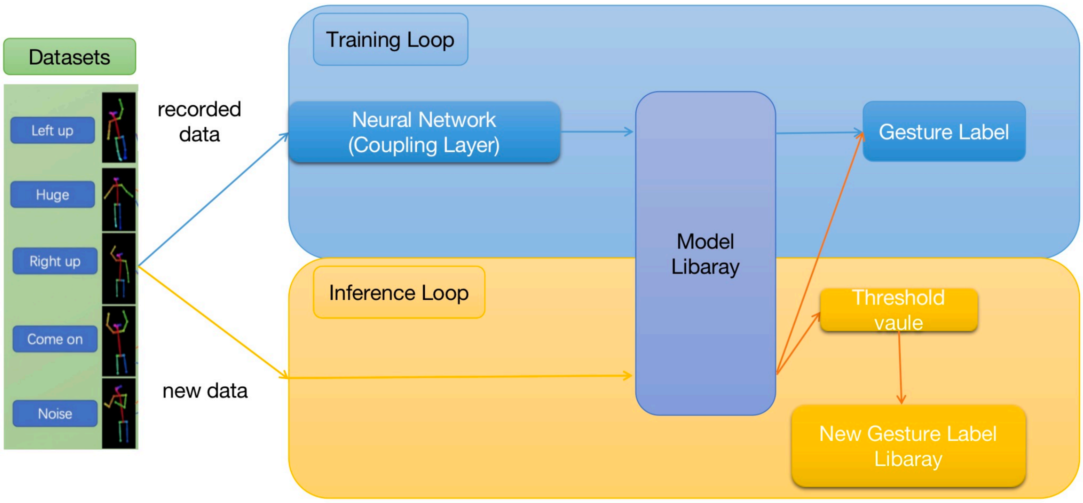


Figure 2.4.: Training and Inference pipeline base on our GestureFlow model.

T-SNE

T-SNE[14] based on [15, 1] is proposed to solve an important problem of visualization of high-dimensional data in many different domains and deals with data of widely varying dimensionality. In Figures 2.5 visualizations of 6,000 handwritten digits from the MNIST dataset is shown with t-SNE. Stochastic Neighbor Embedding (SNE) starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities[1].

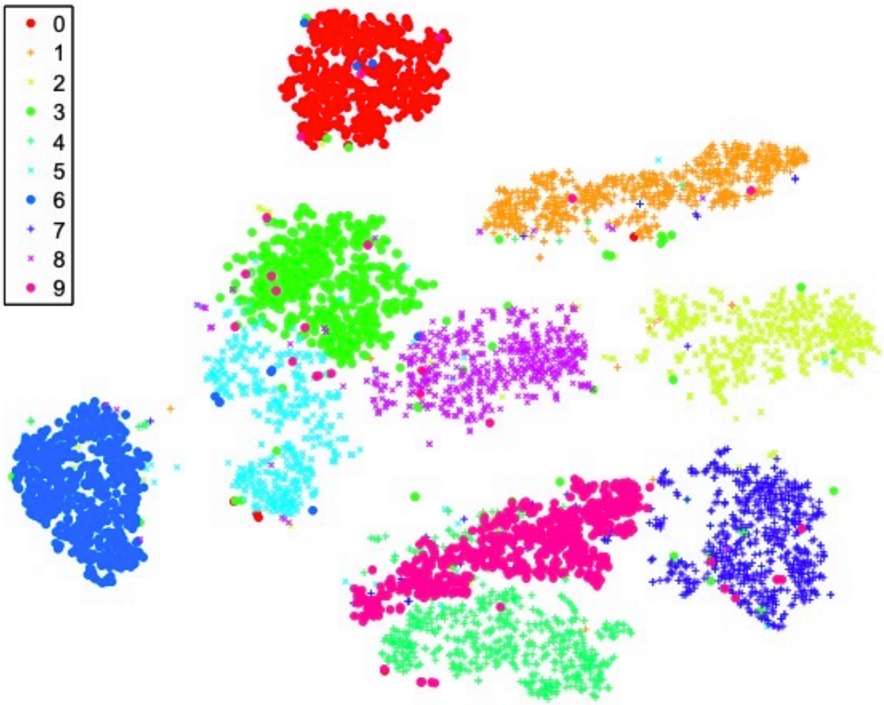


Figure 2.5.: Visualizations of 6,000 handwritten digits from the MNIST dataset[14]

.

The idea behind the T-SNE is minimizing the sum of the Kullback-Leibler divergences between the conditional probabilities $p_{j|i}$ and $q_{j|i}$:

$$KL|P|Q| = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2.26}$$

where P represent probability distribution in the high-dimensional space and Q probability distribution in the low-dimensional space.

$p_{j|i}$ is given by:

$$KL|P|Q| = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq l} exp(-||y_k - y_l||^2)} \tag{2.27}$$

$q_{j|i}$ is defined:

$$KL|P|Q| = \frac{exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{k \neq l} exp(-||x_k - x_l||^2/2\sigma^2)} \tag{2.28}$$

where y_i, y_j, y_k and y_l are the map points, x_i, x_j, x_k and x_l are given datapoints. T-SNE visualizes is helpful for visualization of the distribution of each gesture action in our dataset before we train the data.

2.3. Openpose

Openpose[16, 17, 18, 19] is designed to jointly learn part locations and their association via two branches of the same sequential prediction process. It can realize posture estimation of human body movements, facial expressions, and finger movements. It is suitable for single and multiple people, and has excellent robustness. Many research community applies the OpenPose in many vision and robotics topics such as person re-identification [20], 3D pose estimation [21], 3D human mesh model generation [22], Human-Computer Interaction [23] and GAN-based video retargeting of human faces [24] and bodies [25]. Moreover, OpenPose and PAF-based network architecture [16] ienclosed by Deep Neural Network (DNN) module are comprehended in OpenCV library [26].

2.3.1. Network Architecture

[16] proposes the network architecture, shown in 2.6. In blue block shows the prediction of the affinity field that encodes the association between parts iteratively. In beige block shows detection confidence maps. The iterative prediction architecture, following [27] which improves the prediction of successive stages, $t \in 1, ..., T$, with intermediate supervision at each stage.The network depth has increased relative to [17].

Compared to original approach, [16] reduce the computation of model but remains the receptive field. Due to the 7x7 convolutional layers of original approach [16] utilizes 3 consecutive 3x3 kernels, obviously the number of operations for the former also are decreased. In addition, it concatenates the output of each of the 3 convolution kernels , following an approach similar to DenseNet [28].

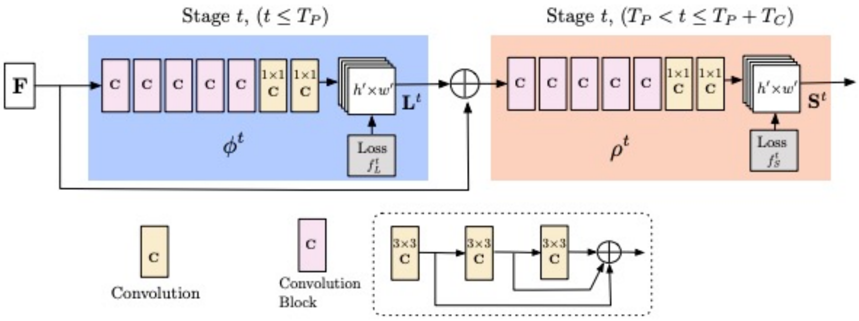
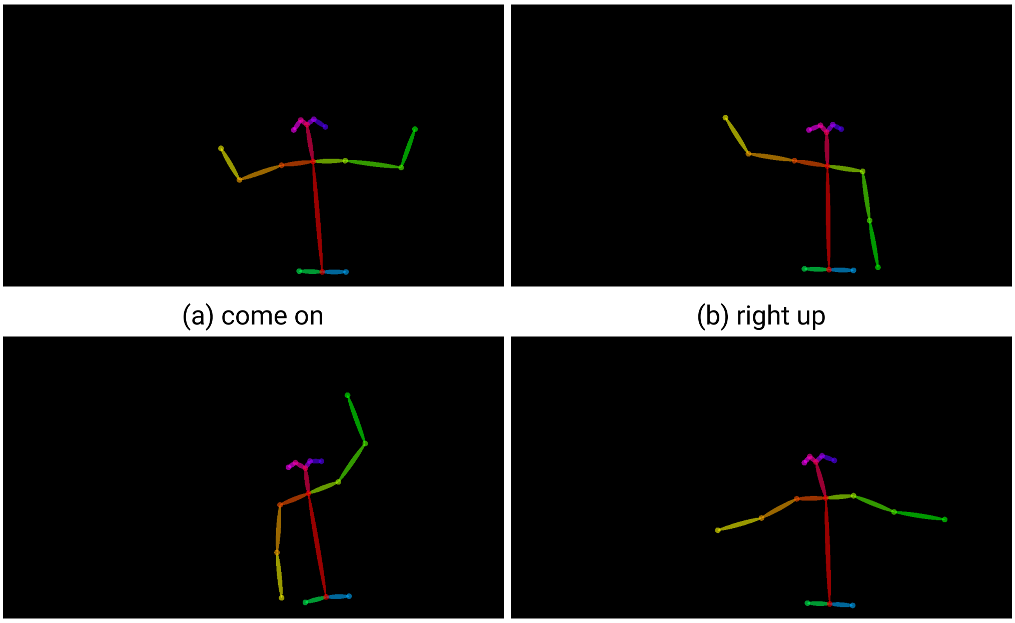


Figure 2.6.: The architecture of the multi-stage CNN consists of the prediction of PAFs L^t and confidence maps S^t. For each subsequent stage, merge the predictions of each stage and its corresponding image features at the same time. Convolutions of kernel size 7 from the original approach [17] are replaced with 3 layers of convolutions of kernel 3 which are concatenated at their end[16].

2.3.2. System

Body and foot detection, hand detection and face detection are important bolcks of OpenPose which can alternatively use the original body-only models [17] trained on COCO and MPII datasets. According to the output of the body detector, the recommended facial bounding box can be roughly estimated from certain parts of the body (especially ears, eyes, nose and neck). Similarly, it uses arm key points to generate hand bounding box suggestions. [19] explain the hand keypoint detector algorithm, while the facial keypoint detector has been trained in the same fashion as that of the hand keypoint detector. The library also performs 3D triangulation by non-linear Levenberg-Marquardt refining the results of multiple synchronized camera views [29], and also includes 3D keypoint pose detection.

Figure 2.7 shows datasets which are collected based on openpose.



(a) come on

(b) right up

(c) left up

(d) huge

Figure 2.7.: 4 types of human skeleton gestures in our datasets, such as come on, right up, left up and left up.

3. Related Work

3.1. Skeleton-based action recognition

Gesture recognition is an ideal example of multidisciplinary research. There are different tools for gesture recognition, based on the approaches ranging from statistical modeling, computer vision and pattern recognition, image processing, connectionist systems, etc. [30]

There are many skeleton-based action recognition based on different probabilistic graphical models have been used for recognizing human gestures and activities. [31] studies on action recognition mainly focus on recognizing actions from RGB videos recorded by 2D cameras (Weinland, Ronfard, and Boyerc 2011). [32] present translation-scale invariant Image mapping and multi-scale deep CNN for human action recognition from 3D skeleton sequences extracted from depth data. For learning feature representations and model long-term temporal dependencies automatically, [33] propose an end to-end fully connected deep LSTM network for skeleton based action recognition. [34] a view adaptive recurrent neural network (RNN) with LSTM architecture, which enables the network itself to adapt to the most suitable observation viewpoints from end to end. [35] pointed out that the method based on the recent recurrent neural network (RNN) mainly focuses on the time evolution of body joints and ignores the geometric relationship. [35] using geometric relationships between joints for motion recognition which propose a novel viewpoint transformation layer and temporal dropout layers are utilized in the RNN based network to learn robust representations. [36] corporate the joint and bone information in skeleton data for action recognition tasks. [36] represent the skeleton data as a directed acyclic graph (DAG) based on the kinematic dependency between the joints and bones in the natural human body, which design directed graph neural network is to extract the information of joints, bones and their relationships and make prediction based on the extracted features. [37] indicate the sparse skeleton information alone is not sufficient to fully characterize human motion that limits makes several existing methods incapable of correctly classifying action categories which exhibit only subtle motion differences. [37] propose a novel framework for employing human pose skeleton and jointcentered light-weight information jointly in a two-stream graph convolutional network, namely, JOLO-GCN. [38] denotes current state-of-the-art methods for action recognition are strongly supervised, i.e., rely on providing labels for training. [38] proposes unsupervised training, the decoder and the encoder self-organize their hidden states into a feature space which clusters similar movements into the same cluster and distinct movements into distant clusters. [39] propose “Music Gesture,” a keypoint-based structured representation to explicitly model the body and finger movements of musicians when they perform music.

3.2. Lifelong Learning

Lifelong learning capabilities are crucial for computational systems and autonomous agents interacting in the real world and processing continuous streams of information.[40]. [41] extends learning without forgetting by preserving important low dimensional feature representations of previous tasks. For each task, an undercomplete autoencoder is optimized end-to-end, projecting features on a lower dimensional manifold. [42] show that sequential fine tuning renders the network unable to properly generate images from previous categories (i.e. forgetting). [42] propose Memory Replay GANs (MeRGANs), a conditional GAN framework that integrates a memory replay generator. [43] introduce a new training strategy, iCaRL, that allows learning in such a classincremental way: only the training data for a small number of classes has to be present at the same time and new classes can be added progressively.

3.3. Normalizing Flow

Whitening transformations [44] transform data into white noise—are the clearest intellectual predecessor to the use of normalizing flows. [45] were perhaps the first to use whitening as a density estimation technique rather than for feature pre-processing, calling the method Gaussianization. [46] studies Gaussization from the perspective of the diffusion process and establish a connection with statistical mechanics-specifically, the Liouville equation is used to describe the flow rate. [47] introduce the modern concept that can be considered as normalizing flow: introduce the term normalized flow and generally define flow as a combination of K simple maps. The idea of composition saw its recent emergence in machine learning starting with [48], who the first to realize that the use of deep neural networks to parameterize flows may result in fairly general and expressive distribution classes. [49] view copulas as rudimentary flows, where Use the empirically estimated marginal cumulative distribution function to independently transform each dimension. Optimal transportation and Wasserstein metric [50] can also be formulated based on the conversion of metric ("measured transportation"). The triangular mapping (a concept closely related to autoregressive flow) can prove to be a limited solution to a class of Monge–Kantorovich problems [51]. Normalizing Flows were popularised by [8] in the context of variational inference which have become a ubiquitous part of modern neural networks [52]. However, the framework is defined in Tabak and VandenEijnden [46]. [53] explores it for clustering and classification, and density estimation [54, 48, 55]. [56] discussing limited and infinitely flows (as we and collating the latest results of density estimation. [**?**]. Affine coupling: Two simple forms of coupling functions were proposed by [8]. Affine coupling functions are used for [57, 58]. [59] proposed an invertible non-linear squared transformation. [60] proposed the Flow++ model, which contains some improvements, including more expressive coupling functions. Variations based on the combination of transformers have been used in the Variations based on the combination of transformers have been used in the following models: NAF [61], block-NAF [62], and Flow++ [60].

4. Experiments

In this chapter, we evaluate and discuss comparison between baseline model and GestureFlow model in 2- and 3-dimensional coordinate and compare their prediction performance with confusion matrix. The summary of the evaluation is concluded in the this chapter.

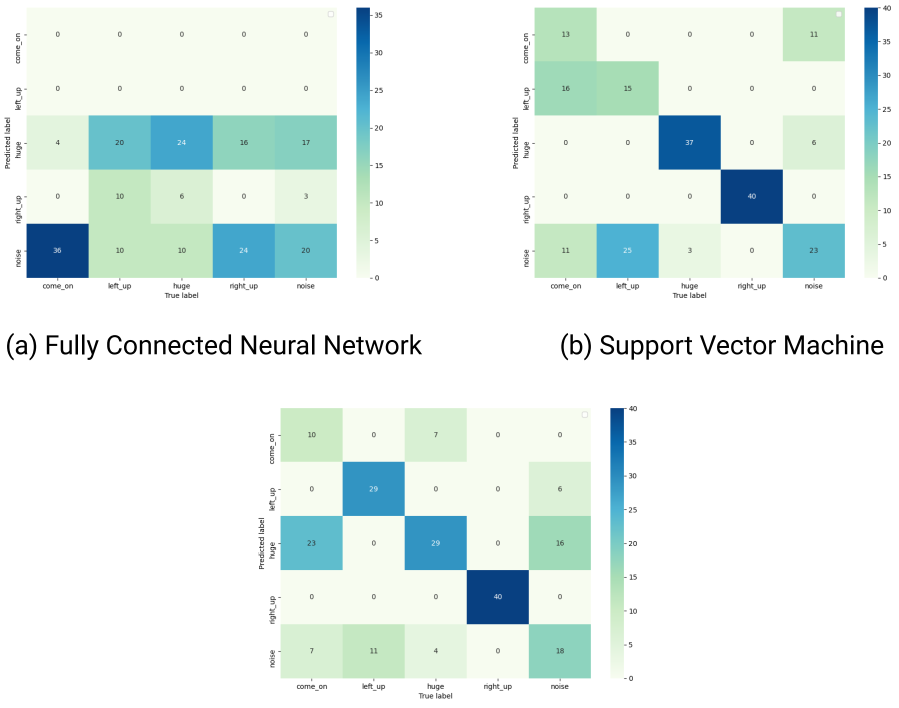
4.1. Comparison GestureFlow model with baseline model in 2 dimensional coordinates

4.1.1. Evaluation on human skeleton data

Based on the openpose framework, we collect 5 types of two-dimensional static actions of human skeleton. The types of our dataset consist of come on, right up, left up, huge and noise(random action) gesture.

In our experiments, our dataset as input contains two types. The two-dimensional absolute coordinate position of the human skeleton is used as input data in training process. Our actions are based on the upper body, the 10 points of the human skeleton are considered as well as the dimension of data is 20. Moreover, human skeleton also are processed as joint angles as input data as well as the dimension of data is 6.

Figure 4.1 shows the the prediction accuracy of the baseline and GestureFlow model based on 40 samples of each type of human skeleton action in absolute coordinate. Figure 4.1a, Figure 4.1b and Figure 4.1c demonstrate the results of prediction which stay in awful situation.



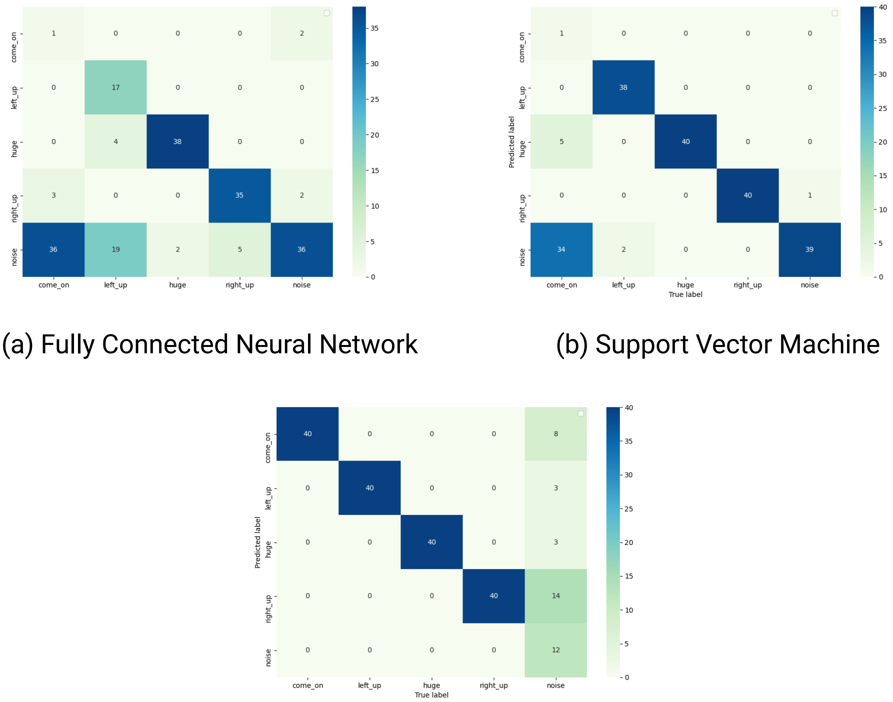
(a) Fully Connected Neural Network

(b) Support Vector Machine

(c) GestureFlow Model

Figure 4.1.: The newly collected skeleton data which are in absolute coordinate. we evaluate datasets in the baseline model(Fully Connected Neural Network and Support Vector Machine) and GestureFlow with confusion matrix.

Therefore, we introduce data preprocessing to convert absolute coordinates into joint angles. Extract the features we need from the data. Figure 4.2 shows the evaluation results for the prediction task on the baseline model and GestureFlow model and the best results are achieved with data preprocessing compare to absolute coordinate. In figure 4.2a the prediction OF FCM achieves good results among datasets right up, huge and noise while it performs worse output among datasets come on and left up. Figure 4.2b shows the good prediction of SVM except the datasets come on. In figure 4.2c our proposed GestureFlow model achieves significantly better scores across considered metrics in confusion matrix and the best recognition of random human gesture actions.



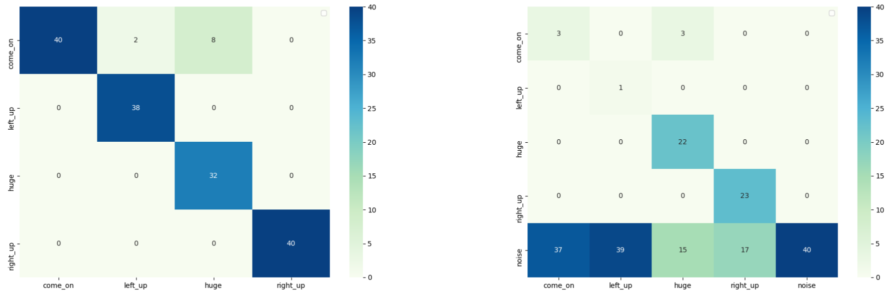
(a) Fully Connected Neural Network

(b) Support Vector Machine

(c) GestureFlow Model

Figure 4.2.: The newly collected skeleton data which are processed into joint angle. we evaluate datasets in the baseline model(Fully Connected Neural Network and Support Vector Machine) and GestureFlow with confusion matrix.

Normalizing flow has a powerful expression function for kernel density estimation. However, in our case, using only Gaussian distribution for maximum likelihood estimation cannot satisfy our demand. Figure 4.2c compares Gaussian Naive Bayes classification for the two different datasets situation. The above evaluation shows that Gaussian Naive classification can successfully make a good prediction without human noise action and indicates that human noise action make our classification less accuracy. The proposed GestureFlow model make our classification more accurate, and the Gaussian distribution cannot be content to solve our problem.

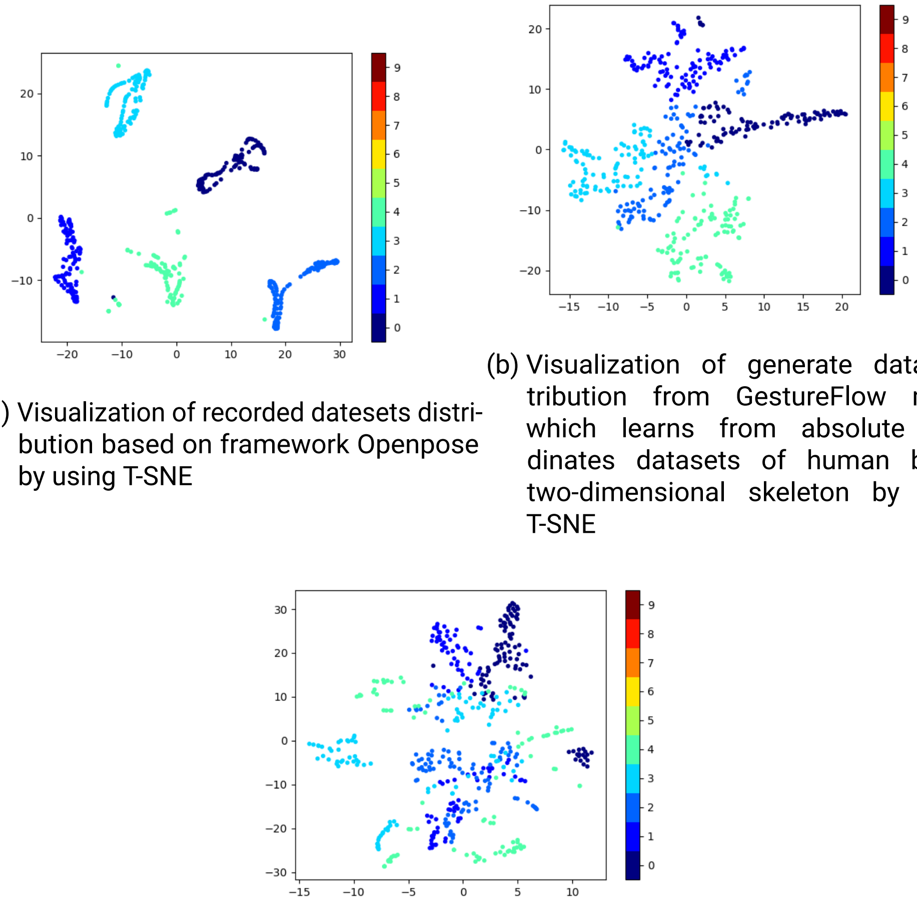


(a) Gaussian Naive Bayes classification without noise datasets

(b) Gaussian naive Bayes classification with noise datasets

Figure 4.3.: The newly collected skeleton data which are processed into joint angle. we evaluate datasets in Gaussian Naive Bayes classification with human noise action and without human noise action.

In order to compare the original data set and the generated data of the model from high-dimensional mapping to low-dimensional distribution. Figure 4.4 illustrates data distribution from different state. Before learning process, figure 4.4a and 4.4b show no connection from hidden state between the distribution of noise action and the distribution of other actions in the visualization state of T-SNE. In figure 4.4c noise action distribution can be related to other action distribution. Therefore, we can use the threshold to infer the labels of other actions without retraining the model.



(a) Visualization of recorded datesets distribution based on framework Openpose by using T-SNE

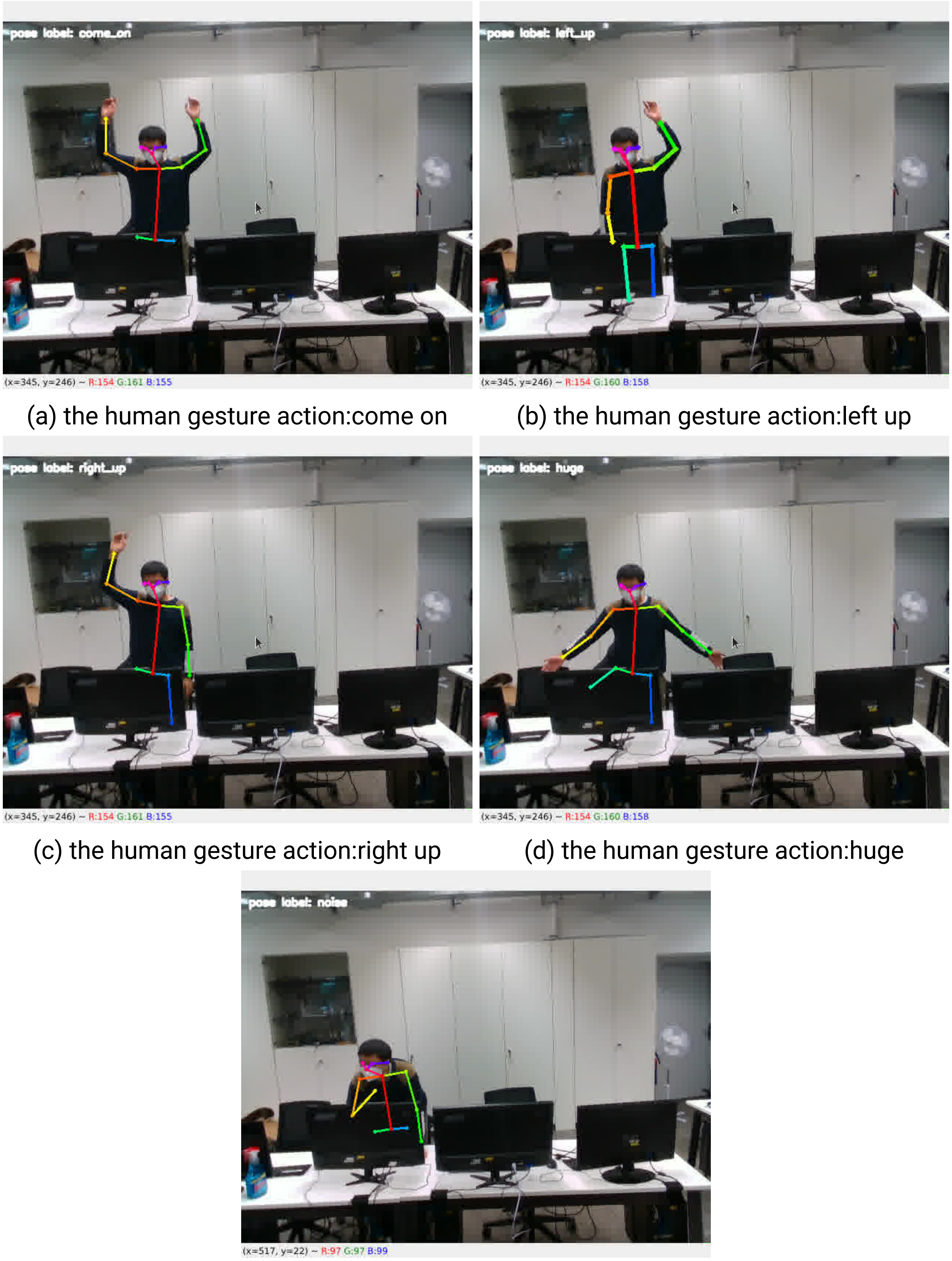
(b) Visualization of generate data distribution from GestureFlow model which learns from absolute coordinates datasets of human body's two-dimensional skeleton by using T-SNE

(c) Visualization of generate data distribution from GestureFlow model which learns from Joints angle of human skeleton datasets of human body's two-dimensional skeleton by using T-SNE

Figure 4.4.: Visualization of original and generate data by using T-SNE(0 represent come on action, 1 represent left up action, 2 represent right up action, 3 represent huge action , 4 represent noise action)

4.1.2. real-time prediction

To evaluate the performance of GestureFlow model in real-time, figure 4.9e shows the real-time results for the GestureFlow model prediction task on human gesture action. Our approaches outperform in real-time. The correct label can be recognized for the same action of human beings in different positions.


pose label: come_on
(x=345, y=246) ~ R:154 G:161 B:155
pose label: left_up
(x=345, y=246) ~ R:154 G:160 B:158
pose label: right_up
(x=345, y=246) ~ R:154 G:161 B:155
pose label: huge
(x=345, y=246) ~ R:154 G:160 B:158
pose label: noise
(x=517, y=22) ~ R:97 G:97 B:99

(a) the human gesture action:come on

(b) the human gesture action:left up

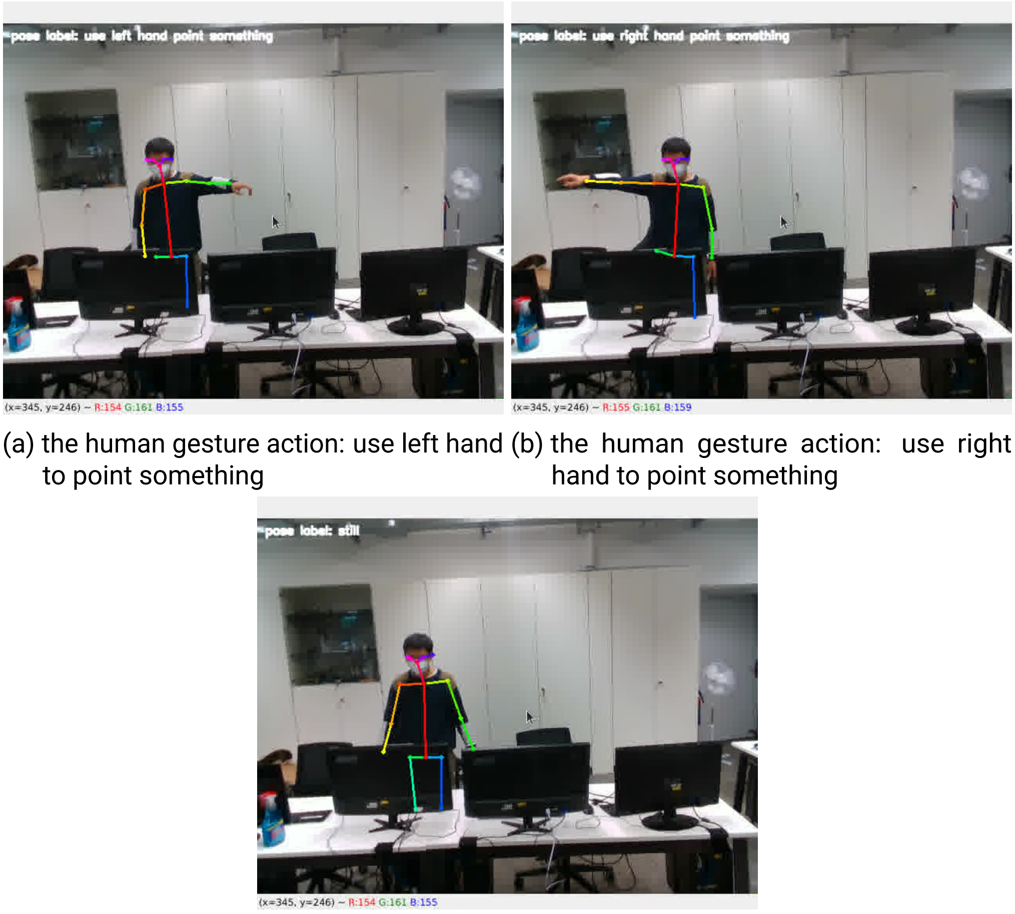
(c) the human gesture action:right up

(d) the human gesture action:huge

(e) the human gesture action:noise

Figure 4.5.: real-time prediction for human gesture action

Our proposed Gesture model can derive the new label for the human gesture action without retraining the model through the threshold. Compared to the baseline model it perform less computational. Otherwise, when baseline is applied in new human gesture action classification, it must be retrained with new label. Figure 4.6 shows the prediction results for new human gesture action. Apparently, it takes a good advantage compared to the baseline model.


pose label: use left hand point something
(x=345, y=246) ~ R:154 G:161 B:155
pose label: use right hand point something
(x=345, y=246) ~ R:155 G:161 B:159
pose label: still
(x=345, y=246) ~ R:154 G:161 B:155

(a) the human gesture action: use left hand to point something

(b) the human gesture action: use right hand to point something

(c) the human gesture action: still

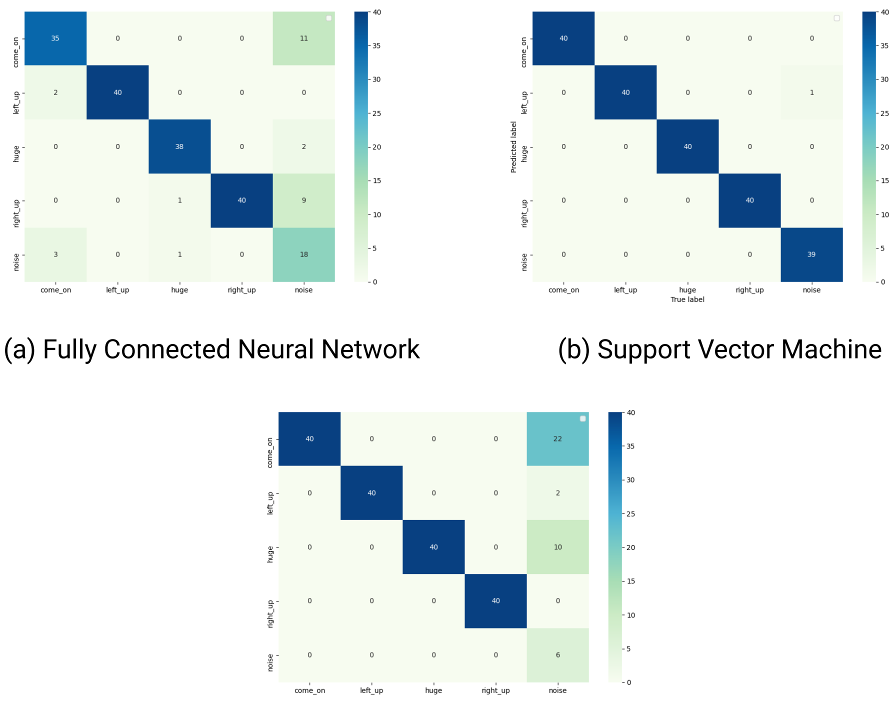
Figure 4.6.: real-time prediction for human new gesture action without retraining the model

4.2. Comparison GestureFlow model with baseline model in 3 dimensional coordinates

We have discussed the evaluation of the prediction results of the human skeleton pose based on the two-dimensional coordinates of the baseline model and the GestureFlow model. Furthermore, we need to evaluate the prediction results of the model human skeleton movement based on three-dimensional coordinates. The three-dimensional absolute coordinate position of the human skeleton are processed as joint angles as input data as well as the dimension of data is 8.

4.2.1. Evaluation on human skeleton data

In figure 4.7 the predictive performance of three models gets evaluated, namely the FCM, SVM and the GestureFlow model, which are learned from preprocessing datasets(joint angles). There is no doubt that the prediction performance of the baseline model is greatly improved when the datasetS is transformed from 2D to 3D, and the gestureflow model still maintains its excellent prediction performance. Our models outperform the other models on this task without retraining the performs better.



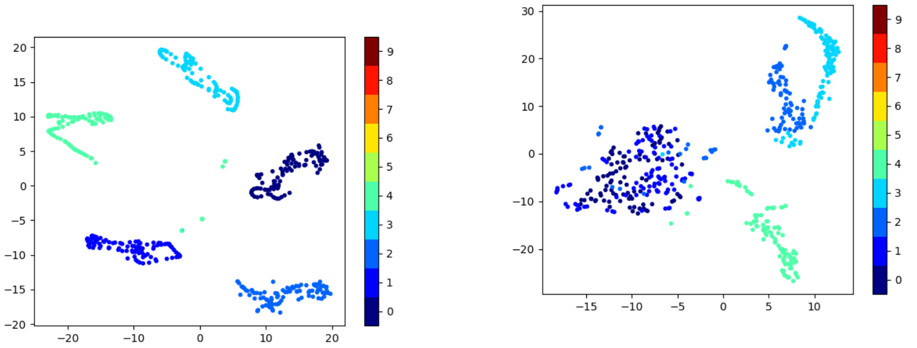
(a) Fully Connected Neural Network

(b) Support Vector Machine

(c) GestureFlow Model

Figure 4.7.: The newly collected skeleton data are evaluated in the confusion matrix of the baseline model(Fully Connected Neural Network and Support Vector Machine) and GestureFlow Model respectively.

In figure 4.8, the visualization of original dataset and the generated data of the model is showed. Compared to 2-dimensional datasets, the distribution of orignal datasets of 3-dimensional remains the same state. Furthermore, in figure 4.8b the model which learns from 3-dimensional joints angle, implies the the hidden state from generate data in each action.



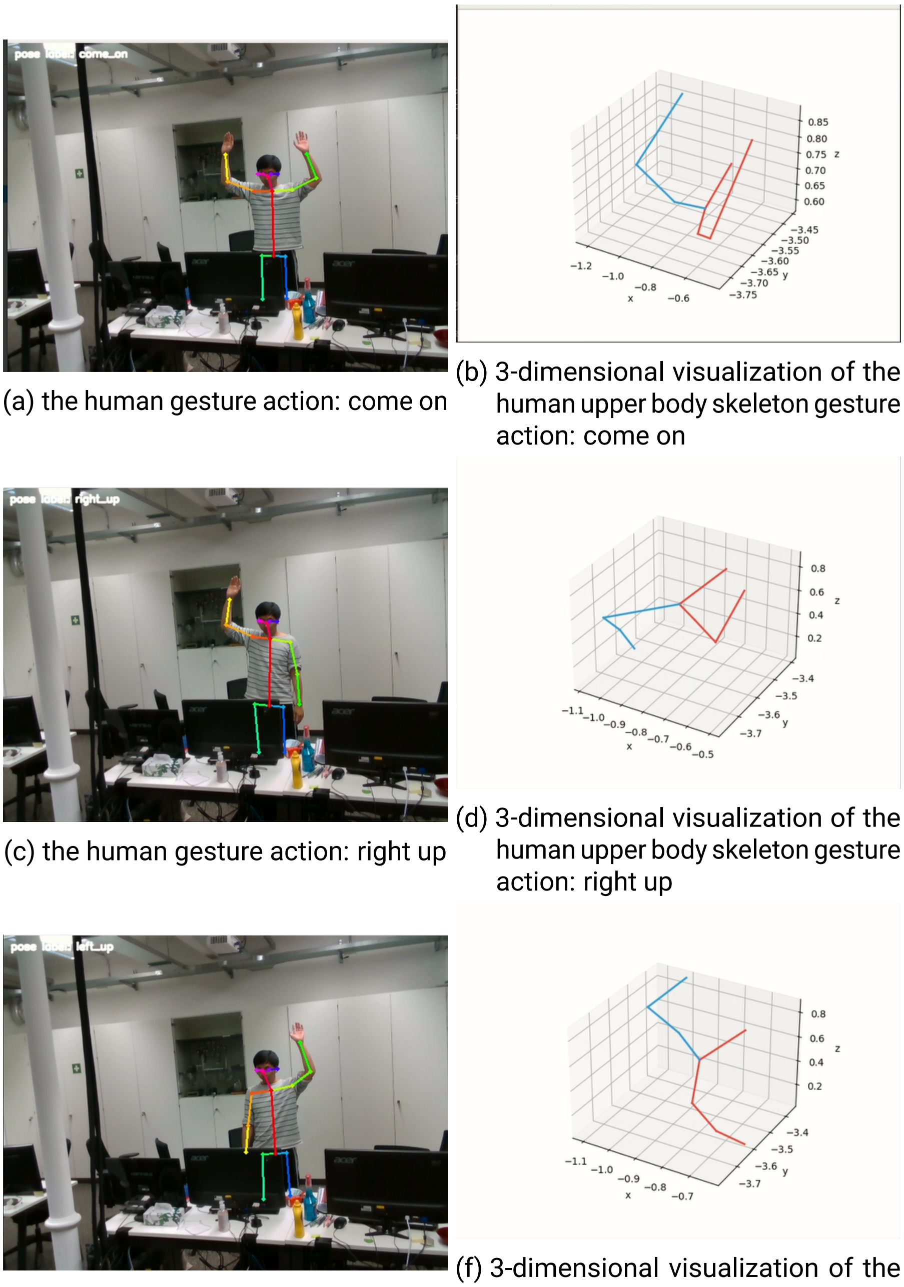
(a) Visualization of recorded datesets distribution based on framework Openpose by using T-SNE

(b) Visualization of generate data distribution from GestureFlow model which learns from joints angle of human skeleton datasets of human body's two-dimensional skeleton by using T-SNE

Figure 4.8.: Visualization of original and generate data distribution by using T-SNE(0 represent come on action, 1 represent left up action, 2 represent right up action, 3 represent huge action , 4 represent noise action)

4.2.2. real-time prediction

To evaluate the performance of GestureFlow model in real-time with 3-dimensional coordinate, figure 4.9 shows the real-time results for the GestureFlow model prediction task on human gesture action and we visualize the human skeleton gesture action in 3-dimensional coordinate. Compared to the real-time prediction of human skeleton gesture in 3-dimensional coordinate, in 2-dimensional coordinate GestureFlow model perform better.



(a) the human gesture action: come on

(b) 3-dimensional visualization of the human upper body skeleton gesture action: come on

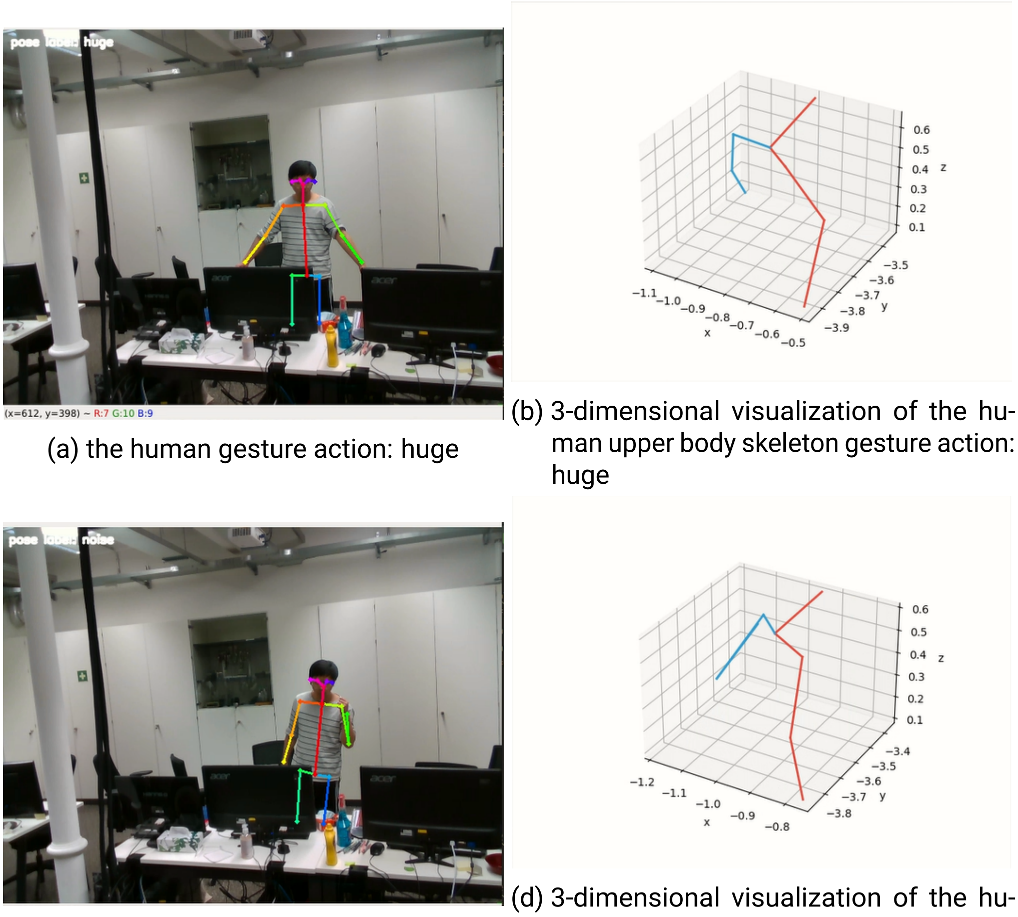
(c) the human gesture action: right up

(d) 3-dimensional visualization of the human upper body skeleton gesture action: right up

(e) the human gesture action: left up

(f) 3-dimensional visualization of the human upper body skeleton gesture action: left up

Figure 4.9.: real-time prediction for human gesture action



(a) the human gesture action: huge

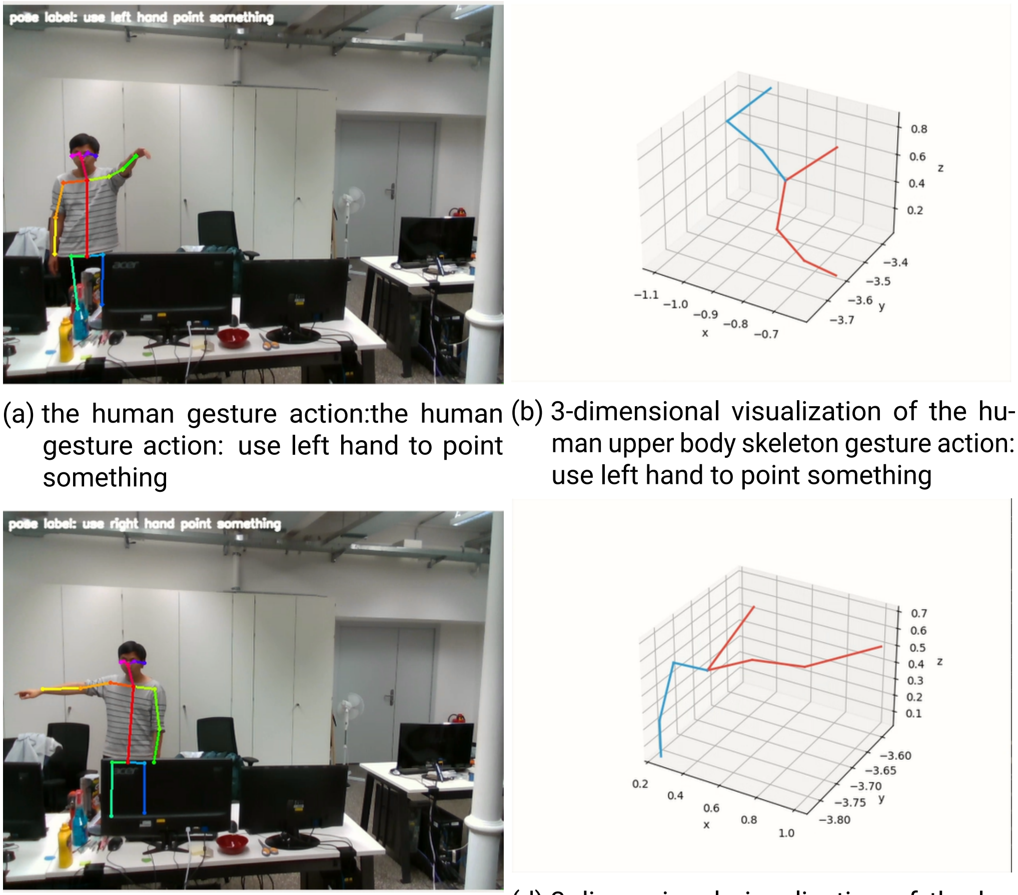
(b) 3-dimensional visualization of the human upper body skeleton gesture action: huge

(c) the human gesture action:noise

(d) 3-dimensional visualization of the human upper body skeleton gesture action: noise

Figure 4.9.: real-time prediction for human gesture action

Our proposed Gesture model can also derive the new label for the human gesture action in 3-dimensional coordinate without retraining the model through the threshold. Figure 4.6 shows the prediction results for new human gesture action.



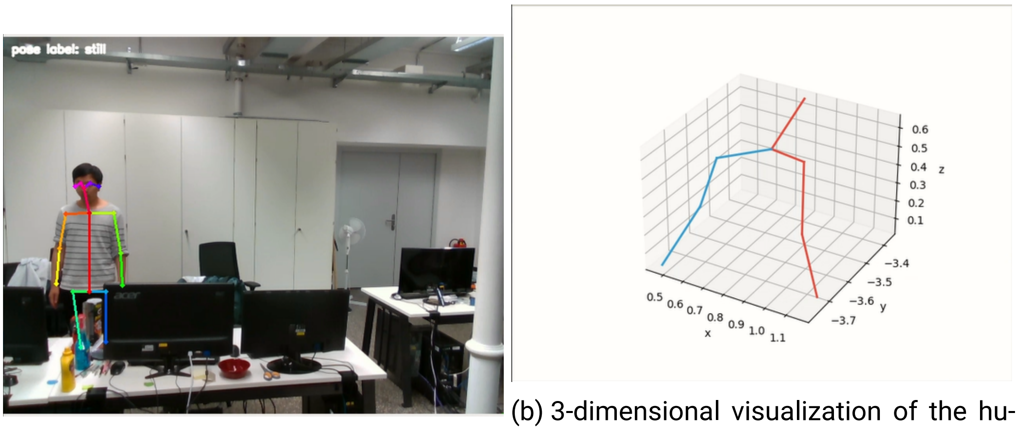
(a) the human gesture action:the human gesture action: use left hand to point something

(b) 3-dimensional visualization of the human upper body skeleton gesture action: use left hand to point something

(c) the human gesture action: use right hand to point something

(d) 3-dimensional visualization of the human upper body skeleton gesture action: use right hand to point something

Figure 4.10.: real-time prediction for human new gesture action without retraining the model



(a) the human gesture action: still

(b) 3-dimensional visualization of the human upper body skeleton gesture action: still

Figure 4.10.: real-time prediction for human new gesture action without retraining the model

5. Results

This thesis aimed to design and investigate an human skeleton gesture action detection based on Normalizing Flow. Inspired by coupling layer, we derived the GestureFlow model to classify the human skeleton gesture action. Then, we compared the GestureFlow model to compare with the baseline model. Based on these models, we studied the effects of the noise action on the others human gesture action. The following benefits of GestureFlow model are listed: Firstly, our proposed method is based on a small number of samples that can obtain new sample labels without retraining the model, it demands less computational compared to Baseline model. Secondly, compared with the baseline model and the prediction accuracy rate is better and perform more stable.

6. Outlook

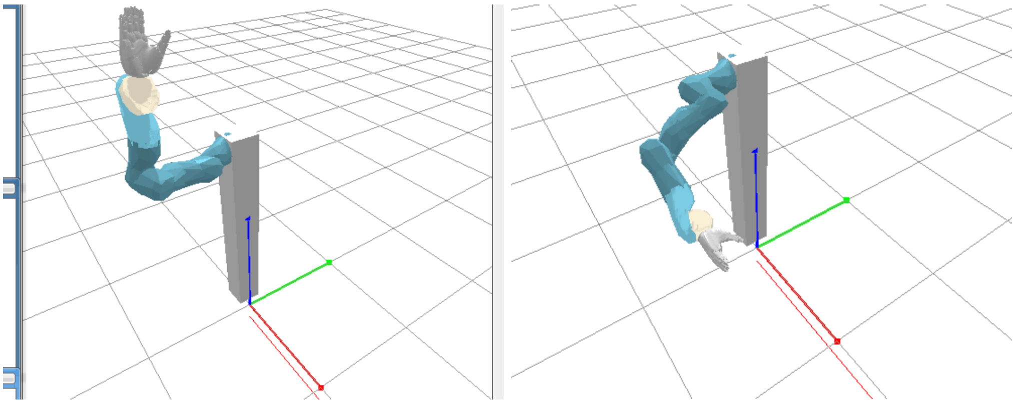
In this study, we achieved significant improvements in human skeleton gesture recognition with Normalizing Flows. On the one hand, if there is not enough information about the current state (including situations related to the high risks posed), GestureFlow can be enabled to help the robot to understand the interaction task. On the other hand, human users are provided with the opportunity to express their preferences.

6.1. Time series human action prediction

Our work does not consider time series human action prediction. [63] extends the Normalizing Flows framework to learn stable Stochastic Differential Equations. In real human-robot interaction, human actions are continuous and related to event. Our work is only based on the recognition of the static movements of the human skeleton. while we need to consider the hidden state of human continuous actions. It makes the models better comparable.

6.2. Human-Robot interaction

In a deeper perspective, the application of GestureFlow is advance improvement in the interaction between robots and humans. Gestureflow recognizes that human actions express human intentions to the robot through the state machine, and the existing action labels help the robot to respond accordingly. Figure 6.1 shows the interactive actions outcome by the robot after recognizing human actions. In essence, we hope and encourage further user research with non-experts to further apply our method.



(a) hello action

(b) shake hand action

Figure 6.1.: Pybullet simulation environment for Human-Robot interaction.

Bibliography

[1] G. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *NIPS*, vol. 15, pp. 833–840, Citeseer, 2002.

[2] C. Cortes and V. Vapnik, “Support vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[3] I. Kobyzev, S. Prince, and M. Brubaker, “Normalizing flows: An introduction and review of current methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[5] H. Zhang, “Exploring conditions for the optimality of naive bayes,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 02, pp. 183–198, 2005.

[6] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *International Conference on Machine Learning*, pp. 1530–1538, PMLR, 2015.

[7] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *arXiv preprint arXiv:1912.02762*, 2019.

[8] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” *arXiv preprint arXiv:1410.8516*, 2014.

[9] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, “Neural spline flows,” *arXiv preprint arXiv:1906.04032*, 2019.

[10] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving variational inference with inverse autoregressive flow," *arXiv preprint arXiv:1606.04934*, 2016.

[11] G. Papamakarios, "Neural density estimation and likelihood-free inference," *arXiv preprint arXiv:1910.13233*, 2019.

[12] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*, pp. 1278–1286, PMLR, 2014.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[15] J. Cook, I. Sutskever, A. Mnih, and G. Hinton, "Visualizing similarity data with a mixture of maps," in *Artificial Intelligence and Statistics*, pp. 67–74, PMLR, 2007.

[16] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[17] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[18] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[19] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[20] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue, "Pose-normalized image generation for person re-identification," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 650–667, 2018.

[21] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 436–445, IEEE, 2018.

[22] H. Joo, T. Simon, and Y. 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*, pp. 8320–8329, 2018.

[23] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. Moura, and M. Veloso, "Teaching robots to predict human motion," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 562–567, IEEE, 2018.

[24] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-gan: Unsupervised video retargeting," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 119–135, 2018.

[25] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5933–5942, 2019.

[26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[27] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4724–4732, 2016.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[30] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[31] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[32] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Twenty-third international joint conference on artificial intelligence*, 2013.

[33] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[34] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2117–2126, 2017.

[35] H. Wang and L. Wang, “Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection,” *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4382–4394, 2018.

[36] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Skeleton-based action recognition with directed graph neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921, 2019.

[37] J. Cai, N. Jiang, X. Han, K. Jia, and J. Lu, “Jolo-gcn: Mining joint-centered light-weight information for skeleton-based action recognition,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2735–2744, 2021.

[38] K. Su, X. Liu, and E. Shlizerman, “Predict & cluster: Unsupervised skeleton based action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9631–9640, 2020.

[39] C. Gan, D. Huang, H. Zhao, J. B. Tenenbaum, and A. Torralba, “Music gesture for visual sound separation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10478–10487, 2020.

[40] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54–71, 2019.

[41] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, “Encoder based lifelong learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328, 2017.

[42] C. Wu, L. Herranz, X. Liu, Y. Wang, J. Van de Weijer, and B. Raducanu, “Memory replay gans: learning to generate images from new categories without forgetting,” *arXiv preprint arXiv:1809.02058*, 2018.

[43] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[44] R. M. Johnson, “The minimal transformation to orthonormality,” *Psychometrika*, vol. 31, no. 1, pp. 61–66, 1966.

[45] P. N. Sabes and M. I. Jordan, “Advances in neural information processing systems,” in *In G. Tesauro & D. Touretzky & T. Leed (Eds.), Advances in Neural Information Processing Systems*, Citeseer, 1995.

[46] E. G. Tabak, E. Vanden-Eijnden, *et al.*, “Density estimation by dual ascent of the log-likelihood,” *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 217–233, 2010.

[47] E. G. Tabak and C. V. Turner, “A family of nonparametric density estimation algorithms,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, 2013.

[48] O. Rippel and R. P. Adams, “High-dimensional probability estimation with deep density models,” *arXiv preprint arXiv:1302.5125*, 2013.

[49] P. Jaworski, F. Durante, and W. K. Härdle, “Copulae in mathematical and quantitative finance,” *Lecture Notes in Statistics-Proceedings. Springer, Heidelberg*, 2013.

[50] C. Villani, *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.

[51] G. Carlier, A. Galichon, and F. Santambrogio, “From knothe’s transport to brenier’s map and a continuation method for optimal transport,” *SIAM Journal on Mathematical Analysis*, vol. 41, no. 6, pp. 2554–2576, 2010.

[52] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[53] J. Agnelli, M. Cadeiras, E. G. Tabak, C. V. Turner, and E. Vanden-Eijnden, “Clustering and classification through normalizing flows in feature space,” *Multiscale Modeling & Simulation*, vol. 8, no. 5, pp. 1784–1802, 2010.

[54] Q. Li, T. Zhang, H. Wang, and Z. Zeng, “Dynamic accessibility mapping using floating car data: a network-constrained density estimation approach,” *Journal of Transport Geography*, vol. 19, no. 3, pp. 379–393, 2011.

[55] L. Baird, D. Smalenberger, and S. Ingkiriwang, “One-step neural network inversion with pdf learning and emulation,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 966–971, IEEE, 2005.

[56] P. Jaini, I. Kobyzev, Y. Yu, and M. Brubaker, “Tails of lipschitz triangular flows,” in *International Conference on Machine Learning*, pp. 4673–4681, PMLR, 2020.

[57] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” *arXiv preprint arXiv:1605.08803*, 2016.

[58] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” *arXiv preprint arXiv:1807.03039*, 2018.

[59] Z. Ziegler and A. Rush, “Latent normalizing flows for discrete sequences,” in *International Conference on Machine Learning*, pp. 7673–7682, PMLR, 2019.

[60] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, “Flow++: Improving flow-based generative models with variational dequantization and architecture design,” in *International Conference on Machine Learning*, pp. 2722–2730, PMLR, 2019.

[61] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, “Neural autoregressive flows,” in *International Conference on Machine Learning*, pp. 2078–2087, PMLR, 2018.

[62] N. De Cao, W. Aziz, and I. Titov, “Block neural autoregressive flow,” in *Uncertainty in Artificial Intelligence*, pp. 1263–1273, PMLR, 2020.

[63] J. Urain, M. Ginesi, D. Tateo, and J. Peters, “Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows,” *arXiv preprint arXiv:2010.13129*, 2020.

A. Some Appendix

Hyperparameter	Value
Dataset size	500
Network Layer Sizes	21
Learning rate	0.01
Batch size	100
Number of optimization epochs per batch	100000
Activation Functions	ReLU
optimizer	adam

Table A.1.: Hyperparameters of GestureFlow used in our experiments

Hyperparameter	Value
Dataset size	500
Network Layer Sizes	2
Learning rate	0.01
Batch size	100
Number of optimization epochs per batch	100000
Activation Functions	ReLU
optimizer	adam

Table A.2.: Hyperparameters of FCM used in our experiments

Hyperparameter	Value
Dataset size	500
Learning rate	0.01
Batch size	100

Table A.3.: Hyperparameters of SVM used in our experiments