applied sciences

MDPI

Article

Ensemble Learning for Skeleton-Based Body Mass Index Classification

Beom Kwon [1] **and Sanghoon Lee** [2,*]

[1] Division of Network Business, Samsung Electronics Company, Ltd., Suwon 16677, Korea; hsm260@yonsei.ac.kr

[2] Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea

* Correspondence: slee@yonsei.ac.kr

Received: 12 September 2020; Accepted: 2 November 2020; Published: 4 November 2020

Abstract: In this study, we performed skeleton-based body mass index (BMI) classification by developing a unique ensemble learning method for human healthcare. Traditionally, anthropometric features, including the average length of each body part and average height, have been utilized for this kind of classification. Average values are generally calculated for all frames because the length of body parts and the subject height vary over time, as a result of the inaccuracy in pose estimation. Thus, traditionally, anthropometric features are measured over a long period. In contrast, we controlled the window used to measure anthropometric features over short/mid/long-term periods. This approach enables our proposed ensemble model to obtain robust and accurate BMI classification results. To produce final results, the proposed ensemble model utilizes multiple k-nearest neighbor classifiers trained using anthropometric features measured over several different time periods. To verify the effectiveness of the proposed model, we evaluated it using a public dataset. The simulation results demonstrate that the proposed model achieves state-of-the-art performance when compared with benchmark methods.

Keywords: anthropometric features; body mass index; class imbalance; ensemble learning; human skeleton

1. Introduction

Over the past several decades, the percentage of the population that is obese has steadily increased. Since obesity can cause various diseases, its prevalence among people has become a major social issue. To identify its adverse effects, many studies have investigated the relationship between obesity and various diseases. According to the results presented in the literature, obesity is related to several diseases such as diabetes [1], high blood pressure [2], hyperlipidemia [3], cholelithiasis [4], hypopnea [5], arthritis [6] and mental disorders [7], meaning that the death rate for the obese is relatively high.

The World Health Organization (WHO) defines obesity as excessive fat accumulation that may threaten life. Additionally, the WHO uses the body mass index (BMI) to identify obesity in people. The BMI is one of the factors used to estimate obesity levels and is defined as the ratio of a person's body weight in kilograms to the square of their height in meters (i.e., the unit of the BMI is $\mathrm{kg/m^2}$). In general, a person is classified as "overweight" when their BMI is greater than 25.

Traditionally, if a person wants to know their BMI, they must first measure their body weight and height using a weight scale and extensometer, respectively. After this, their BMI can be calculated according to the definition of the BMI above. If no device is used to measure body weight or height, it is impossible to calculate the BMI. To make the acquisition of BMIs simpler, i.e., possible without using such devices, many researchers in the field of computer vision have explored various methods of

BMI prediction/classification. Most have focused on calculating the BMI of an individual from a facial image [8–10]. In the literature, face image databases built under a controlled environment, such as frontal view, clean background, alignment according to interpupillary distance, etc., have been used. However, in a real-world situation, facial images are generally captured in uncontrolled conditions. For example, the facial image of a person can be taken from the side view as well as the frontal view and can have complex backgrounds. In addition, in some cases, a person can wear a mask which conceals parts of their face. In particular, since most of the features used in [8–10] can be extracted from only front-view face images without any occlusions, the feasibility and effectiveness of these methods for real-world application still need to be demonstrated.

Some studies on BMI classification have been conducted using human full-body images [11–13]. In these studies, each individual was asked to stand up straight while their full-body image was taken. In addition, the image was captured from a frontal view. Most of the features proposed in [11–13] can be obtained from only frontal full-body images. Therefore, in a real-world situation, if the methods proposed in [11–13] are used for BMI classification, it is necessary to have the cooperation of an individual to enable the extraction of features from their frontal full-body images.

Recently, based on advancements in human pose estimation techniques, the acquisition of three-dimensional (3D) human motion data has become quicker, simpler and more precise [14–17]. Therefore, many studies have exploited human motion data for various applications. Human action recognition [18–21], animation generation [22–25] and gait recognition [26,27] are prime examples of such research. Compared with conventional facial and full-body image-based BMI classifications, skeleton-based BMI classification has the advantage of being performed without the user's cooperation, with the user only being asked to walk in a usual way. Because the user's gait data can be obtained from the video filmed from various viewpoints as well as a frontal view, the BMI classification process is performed in such a way that the user is unaware of it and does not feel uncomfortable. However, to the best of our knowledge, there has only been one study [28] to date in which a method for classifying BMIs using 3D human motion sequences has been investigated. According to the results presented in [28], classification accuracy was improved up to only 40%. Motivated by these results, in this study, we investigate a method that facilitates the improvement of the classification accuracy of BMI. A block diagram of the proposed BMI classification method is presented in Figure 1.

The main contributions of this study can be summarized as follows.

- We determine why the BMI classification accuracy in [28] was poor. Because the dataset considered had a class imbalance problem, the authors of [28] used the undersampling technique to balance classes. During our investigation, it was determined that the reason for performance degradation stemmed from the use of this technique. Therefore, we found that it is better to use an oversampling technique rather than an undersampling technique in terms of performance.
- We demonstrate the usefulness of anthropometric features in the BMI classification problem. During this study, we observe that feature values calculated for each frame are not consistent over time, as a result of the inaccuracy in pose estimation. Furthermore, for some motion sequences, a large variance in feature values was observed over specific periods. Therefore, instead of using traditional anthropometric features, we employ anthropometric features measured over several different time periods (i.e., long/mid/short-term periods). The use of these features contributes to our ensemble model being able to obtain robust and accurate classification results.
- We propose an ensemble model capable of classifying BMIs from 3D human motion sequences. In the proposed model, multiple k-nearest neighbor (k-NN) classifiers are trained separately using anthropometric features calculated from different time windows. Generalized results are then obtained by integrating the classification results obtained from the k-nearest neighbor (k-NN) classifiers. The evaluation results for the dataset considered in [28] demonstrate that the proposed model achieves state-of-the-art performance.

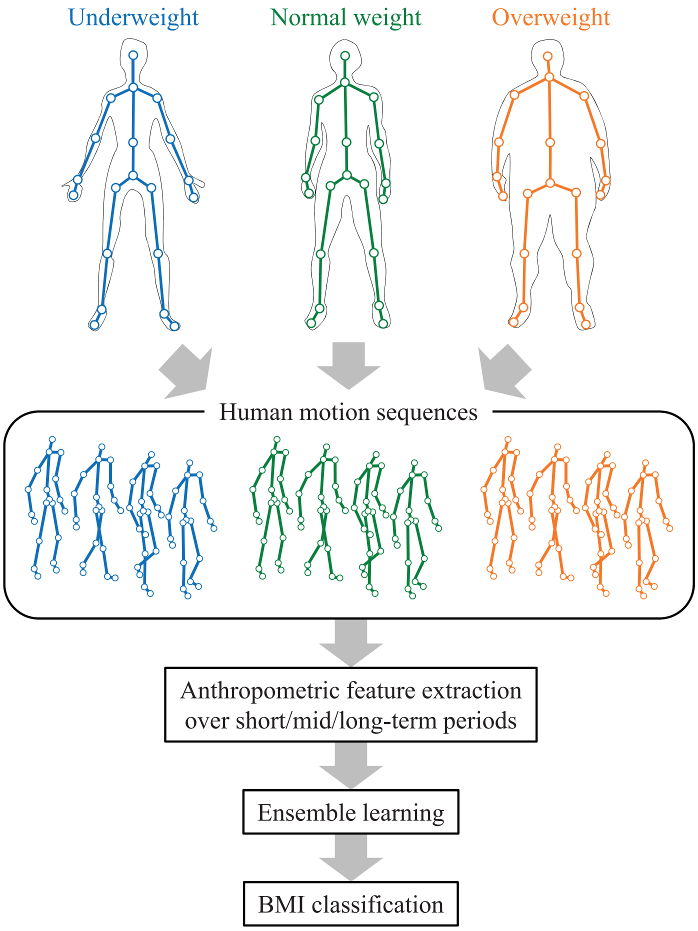


Figure 1. Block diagram for classifying the body mass index (BMI) from 3D human motion sequence.

The remainder of this paper is organized as follows. Section 2 presents a review of related work. The motivation for our study is discussed in Section 3. Our methodology for classifying BMIs from 3D human motion sequences is described in Section 4. The experimental setup is described in Section 5. The results and discussion are presented in Sections 6 and 7, respectively. Conclusions are presented in Section 8.

2. Related Work

Previous studies on BMI classification have mainly focused on calculating human weight or BMI from facial images [8–10,29]. For example, Coetzee et al. [8] demonstrated that three facial features, namely the width-to-height ratio, perimeter-to-area ratio and cheek-to-jaw-width ratio, are highly related to both human weight and BMI. Additionally, they proposed the use of these three features to judge the body weight and BMI of an individual. Motivated by this study, Pham et al. [9] assessed the relationship between these facial features and BMIs for facial images of 911 subjects. They demonstrated that the eye size and average eyebrow height are also related to the BMI. Wen and Guo [10] investigated a method for predicting the BMIs of people from face images. To achieve this goal, they extracted the facial features proposed in [8,9] from passport-style frontal face images with clean backgrounds. They then trained a support vector regression (SVR) model using these features and demonstrated the feasibility of using SVR to predict BMIs. Kocabey et al. [29] explored a method for predicting the BMI using low-quality facial images collected from social media. To achieve this goal, they used the convolutional neural network (CNN) proposed in [30]. They then used the outputs from the sixth fully connected layer of the CNN as features for the BMI prediction. They trained an SVR model using these features and evaluated their model using a publicly available social media face dataset. The evaluation results demonstrated the feasibility of BMI prediction using low-quality facial images.

BMI prediction/classification based on full-body person images has also been studied [11–13]. For example, Bipembi et al. [11] proposed a method for obtaining the BMI of an individual from their full-body silhouette image. To achieve this goal, the authors captured frontal full-body images of standing individuals. They then converted these images into silhouette images. Based on a silhouette image, they calculated the area and height of an individual. The area was defined as the number of pixels occupied by a silhouette, and the height was defined as the distance between the highest and lowest pixels in a silhouette. Finally, the BMI was calculated using the extracted values of the area and height. Amador et al. [12] also used human silhouette images for BMI classification. A total of 57 shape features introduced in [31] were extracted from images. Four machine learning algorithms, namely logistic regression, a Bayesian classifier, an artificial neural network and support vector machine (SVM), were tested on their own dataset consisting of 122 subjects. According to their results, the SVM outperformed the other algorithms and achieved an accuracy of approximately 72.16%. Madariaga et al. [13] proposed the calculation of human height from full-body images using the vanishing point and camera height information. For body weight calculation, a sensor called the Wheatstone bridge load cell was used. Using the estimated height and measured body weight information, the BMI was calculated.

Nahavandi et al. [32] investigated a method for estimating BMIs from depth images of people. To achieve this goal, the deep residual network (ResNet) proposed in [33] was used. To train the ResNet, 3D human polygon models were generated using the MakeHuman open-source software. The body surface area (BSA) of each model was calculated. By using the formula $\mathrm{BSA} = 1/6(\text{body weight} \times \text{height})^{0.5}$ [34], the body weight of each model was estimated. For each polygon model, a BMI was then obtained using the height and estimated body weight. These were then used as ground truth BMI values to train the ResNet. Additionally, 3D depth images were obtained from 3D human polygon models using a rendering software and encoded using red–green–blue (RGB) colorization. These colorized depth images were then used as training data for the ResNet. The authors tested their proposed method on their own dataset. According to their results, the ResNet achieved an accuracy of 95%.

The concept of using anthropometric features for BMI classification was first proposed by Andersson et al. [28]. For BMI classification, they used 20 anthropometric features, including the average length of each limb and the average height of the 3D human skeleton. To test the performance of their model, they constructed a dataset consisting of 112 subjects. Three classes of "underweight", "normal weight" and "overweight" were used to classify BMIs. However, in their dataset, the number of people categorized as underweight was too small. Most people were categorized as normal weight. It is well known that if a classifier is trained using a dataset with an unequal distribution of classes, that classifier will exhibit poor classification performance for minority classes [35–37]. Therefore, in their experiments, Andersson et al. used an undersampling technique that randomly removed some samples from the majority classes to alleviate the imbalance in the dataset. They applied three machine learning algorithms, namely the SVM, *k*-NN model and multi-layer perceptron (MLP), to the undersampled dataset. According to their results, the BMI classification accuracies of the SVM, *k*-NN model and MLP were 40%, 25% and 35%, respectively. Because the accuracies of all algorithms were poor, the authors concluded that the anthropometric features were unhelpful for classifying BMIs.

3. Motivation

According to the results of some studies [38–40], undersampling techniques do not provide a good solution to the class imbalance problem because they may remove some important and useful data from majority classes. This loss of data can result in the performance degradation of a classifier for both majority and minority classes. Therefore, it can be concluded that the performance drop in terms of classification accuracy [28] may stem from the use of undersampling, rather than the unreliability of anthropometric features. To solve this issue, we generated a training dataset by employing the synthetic minority oversampling technique (SMOTE) proposed in [41] to overcome the lack of minority samples. Additionally, for comparison, we generated a training dataset by employing the undersampling technique used in [28].

Figure 2 illustrates the change of the number of datasets according to the used sampling technique. As shown in Figure 2, the number of samples belonging to the underweight class is significantly lower than the numbers of samples belonging to the other two classes. In this figure, one can see that the original dataset has a class imbalance problem. To overcome this problem, the undersampling technique was employed. Accordingly, some samples were removed from the majority classes. As a result, as shown Figure 2, the normal weight and overweight classes have the same number of samples as the underweight class. Although the class imbalance problem is alleviated by the undersampling technique, the total number of samples in the undersampled training dataset is significantly reduced compared to that in the original dataset. Instead of removing some samples from majority classes, SMOTE augments the samples of minority classes by generating new synthetic samples. As a result, as shown in Figure 2, the number of samples in each minority class is the same as that in the majority class.

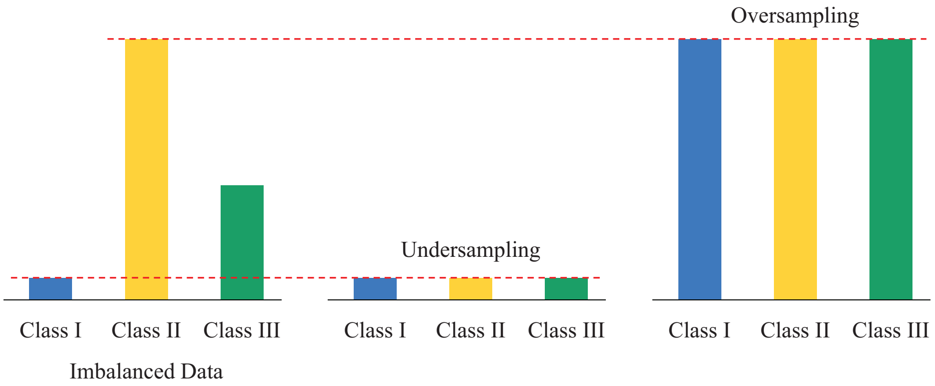


Figure 2. Change of the number of datasets according to sampling technique. Here, the underweight, normal weight and overweight classes are denoted as Classes I, II and III, respectively.

Using the training dataset generated by SMOTE, we trained three machine learning algorithms: C-SVM, nu-SVM and *k*-NN. We then tested these algorithms on the testing dataset. It is noteworthy that the performance of average BMI classification for all three schemes improved significantly. *k*-NN achieved an accuracy of 92.97%, outperforming the other algorithms.

Therefore, we can confirm that anthropometric features hold validity for the classification of BMIs. Next, we investigated a method for improving BMI classification accuracy. To this end, we carefully reviewed the generation process for anthropometric features and measured the effects of each feature on classification. Traditionally, anthropometric features have been calculated using the length of each limb and the height of an individual averaged over the given frames. Figure 3 presents the body height of an individual over multiple frames (the height of the fourth subject in the dataset proposed by Andersson et al. [28]). As shown in this figure, the values are not consistent between frames as a result of the inaccuracy in pose estimation. Furthermore, there is a specific period during which a large variance among the values can be observed.

Therefore, if anthropometric features are simply extracted, ignoring such large variance over time, BMI classification accuracy could be significantly reduced. To overcome this problem, it is desirable to use anthropometric features localized over several different time periods rather than using features averaged over all frames. Therefore, in this paper, we propose an ensemble model for BMI classification. In the next section, we discuss our methodology in detail.

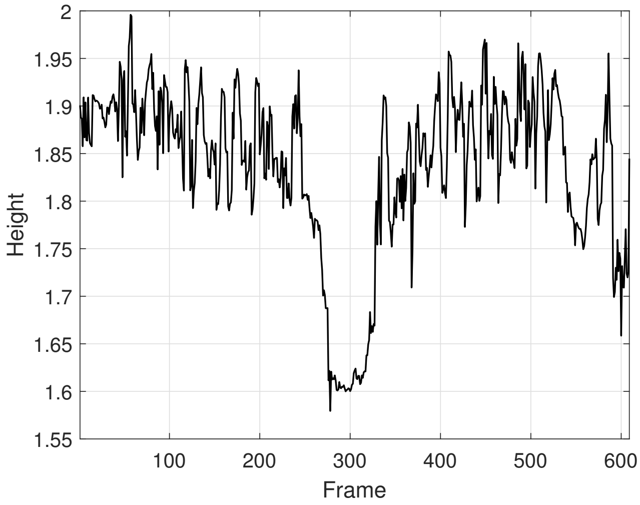


Figure 3. Body heights calculated at each frame.

4. Ensemble for BMI Classification

Figure 4 presents a schematic overview of the proposed method for skeleton-based BMI classification. One can see that a 3D human skeleton sequence is used as an input and is divided into several segments of equal length. In this study, we divide an original sequence into T different segments. Specifically, we let W be the total length of the input motion sequence. In the frame division process, $T \cdot (1+T)/2$ segments corresponding to each length are generated (one for W, two for $W/2$, three for $W/3$, four for $W/4$, five for $W/5$, $\cdots$, T for W/T).

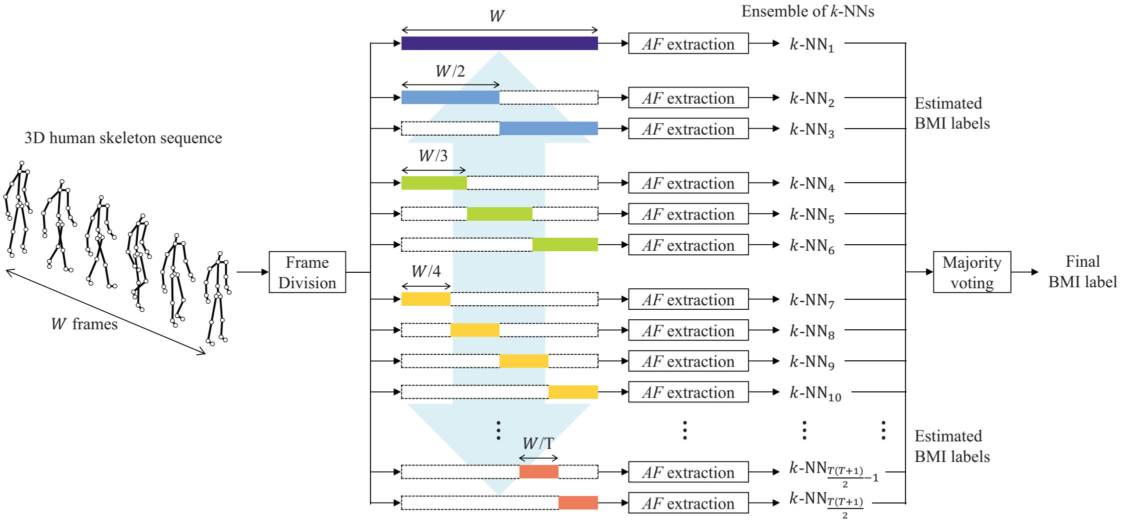


Figure 4. Schematic overview of the proposed method for skeleton-based BMI classification.

For each segment described above, anthropometric features are calculated. For clarification, suppose that a 3D human skeleton model consists of N joints and M limbs. Let $J = \{1, \cdots, N\}$ be the set of joint indices and $L = \{(i,j) | i \neq j, i \in J, j \in J\}$ be the set of joint pairs for constructing limbs. The values of N and M depend on the product specifications of motion capture sensors. For the Microsoft Kinect sensor [42], N and M are 20 and 19, respectively, as shown in Figure 5.

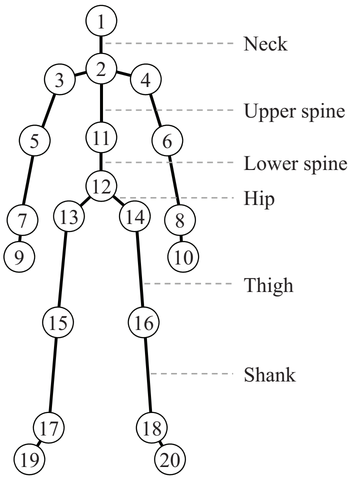


Figure 5. Three-dimensional human skeleton model consisting of 20 joints and 19 limbs.

Let $\mathbf{P}_j[k] = (x_j[k], y_j[k], z_j[k])$ be the position of the jth joint at the kth frame in the world coordinate system, where $x_j[k]$, $y_j[k]$, and $z_j[k]$ are the coordinates of the jth joint at the kth frame on the X, Y, and Z axes, respectively. The Euclidean distance between the ith and jth joints at the kth frame is then calculated as

$$d(\mathbf{P}_i[k], \mathbf{P}_j[k]) = \sqrt{(x_i[k]-x_j[k])^2+(y_i[k]-y_j[k])^2+(z_i[k]-z_j[k])^2}. \tag{1}$$

In [43–50], the average length of each limb and average height of the target individual were used as anthropometric features. In this study, we used these features for BMI classification. To obtain anthropometric features for each segment, the length of each limb and height of the individual are measured in each frame. Average values are then used as the anthropometric features for the corresponding segment.

For clarity, let $F \in \{W, W/2, W/3, \cdots, W/(T-1), W/T\}$ be the frame length of a segmented sequence. The average length of each limb over F is then calculated as

$$AF_{i,j} = \frac{1}{F}\sum_{k=1}^{F} d(\mathbf{P}_i[k], \mathbf{P}_j[k]), \ \ (i,j) \in L. \tag{2}$$

For simplicity, we define a feature vector AF_1 as

$$AF_1 = \left[AF_{i,j} \ \text{ for } \ (i,j) \ \text{ in } \ L\right]. \tag{3}$$

Because we consider the 19 limbs presented in Figure 5, the dimension of AF_1 in (3) is 19.

In general, human height is measured as the distance from the bottom of the feet to the top of the head in a human body. According to this definition of human height, the height of a 3D human skeleton also can be calculated as the distance from the joint of feet to the joint of head. However, to measure the height using this traditional method, it is necessary for the user to stand straight up during height measurement. If the user moves during height measurement, the measured height may be inaccurate. In [43], Araujo et al. proposed a method that is capable of estimating the height of the person even if the person moves. This method is widely used to estimate the height of human skeletons [44–50]. In this study, we used the method to estimate the skeleton's height. According to the method proposed in [43], the target individual's height is calculated as the sum of their neck length, upper and lower spine lengths and average lengths of the right and left hips, thighs and shanks. For clarity, let $H[k]$ be the subject's height at the kth frame. Then, $H[k]$ is defined as

$$
\begin{aligned}
H[k] = \ & \underbrace{d(\mathbf{P}_1[k],\mathbf{P}_2[k])}_{\text{= length of neck}} + \underbrace{d(\mathbf{P}_2[k],\mathbf{P}_{11}[k])}_{\text{= length of upper spine}} + \underbrace{d(\mathbf{P}_{11}[k],\mathbf{P}_{12}[k])}_{\text{= length of lower spine}} \\
& + \underbrace{\{d(\mathbf{P}_{12}[k],\mathbf{P}_{13}[k]) + d(\mathbf{P}_{12}[k],\mathbf{P}_{14}[k])\}/2}_{\text{= average length of the right and left hips}} + \underbrace{\{d(\mathbf{P}_{13}[k],\mathbf{P}_{15}[k]) + d(\mathbf{P}_{14}[k],\mathbf{P}_{16}[k])\}/2}_{\text{= average length of the right and left thighs}} \\
& + \underbrace{\{d(\mathbf{P}_{15}[k],\mathbf{P}_{17}[k]) + d(\mathbf{P}_{16}[k],\mathbf{P}_{18}[k])\}/2}_{\text{= average length of the right and left shanks}}. \qquad (4)
\end{aligned}
$$

By using $H[k]$ from (4), the average height over F is obtained as

$$AF_2 = \frac{1}{F}\sum_{k=1}^{F} H[k]. \qquad (5)$$

After the anthropometric features are obtained using (2) and (5), these features are concatenated into a feature vector as

$$AF = \text{concat}\left(AF_1, AF_2\right), \qquad (6)$$

where $\text{concat}(AF_1, AF_2)$ is an operator that concatenates AF_2 onto the end of AF_1. As a result, the dimension of AF in (6) is 20.

We present the pseudo-code for the proposed ensemble learning algorithm in Algorithm 1. As shown in Figure 4, in the proposed method, the ensemble model consists of multiple (i.e., $T \cdot (T+1)/2$) k-NNs. For simplicity, hereafter, we refer to the $T \cdot (T+1)/2$ k-NNs as k-NN$_1$, k-NN$_2$, k-NN$_3$, $\cdots$, k-NN$_{T\cdot(T+1)/2-1}$, and k-NN$_{T\cdot(T+1)/2}$. In the training (testing) phase, each AF extracted from the $T \cdot (T+1)/2$ sequence segments is inputted into the corresponding k-NN. This can be explained as follows. As shown in Figure 3, the lengths of body parts and the height of the target individual vary over time as a result of the inaccuracy in pose estimation. Since, traditionally, the average values of anthropometric features over all frames (i.e., W) are calculated, the performance of classifiers trained using them can be affected by the variance of features.

In contrast, in the proposed ensemble learning method, anthropometric feature vectors are calculated as averages over long/mid/short-term periods (i.e., W, $W/2$, $W/3$, $\cdots$, W/T). Through the frame division process, the periods in which the variance exists are divided into sub-periods. As a result, the variance for each of the $T \cdot (T+1)/2$ segments is reduced. Additionally, the AFs calculated for $T \cdot (T+1)/2$ segments are used to train the k-NNs in the ensemble model. The use of such AFs helps each k-NN to obtain robust and accurate classification results (i.e., BMI classes).

Let $Q = \{1, 2, 3, \cdots, T \cdot (T+1)/2 - 1, T \cdot (T+1)/2\}$ be the set of indices of k-NNs and q be the index of a k-NN (i.e., $q \in Q$). Additionally, let C_q be the classification result of k-NN$_q$. For simplicity, we define a classification result vector C_Q as

$$
\begin{aligned}
C_Q &= \left[C_q \ \text{ for } \ q \ \text{ in } \ Q\right] \\
&= \left[C_1, C_2, \cdots, C_{T\cdot(T+1)/2-1}, C_{T\cdot(T+1)/2}\right]. \qquad (7)
\end{aligned}
$$

To derive a final classification result from C_Q in (7), in this study, the majority voting algorithm was adopted. The pseudo-code for the majority voting algorithm is presented in Algorithm 2.

Algorithm 1 Pseudo-code for the proposed ensemble learning algorithm (training phase).

```
Input: Training data $D_{\text{train}}$ consisting of $N_{\text{train}}$ human motion sequences
Output: Trained ensemble model $E$
Set the frame division parameter $T$;
Create a 3D matrix of zeros $M_{3d}$;
for seq_idx = 1:1:$N_{\text{train}}$ do
    seq = $D_{\text{train}}$[seq_idx];
    len = size(seq);
    Initialize the index of $k$-NN $q = 0$;
    for div_idx = 1:1:$T$ do
        for seg_idx = 1:1:div_idx do
            $q = q + 1$;
            stard_idx = round(len*(seg_idx-1)/div_idx)+1;
            end_idx = round(len*(seg_idx)/div_idx);
            seg_range = start_idx:1:end_idx;
            segemented_data = $D_{\text{train}}$[seg_range];
            Extract $AF$ from segmented_data using (6);
            $M_{3d}$[seq_idx, 1 : 20, $q$] = $AF$;
            $M_{3d}$[seq_idx, 21, $q$] = corresponding_BMI_class;
        end for
    end for
end for
model = build_model($T$);
# In build_model, an ensemble model consisting of $T \cdot (T+1)/2$ $k$-NNs are constructed.
$E$ = ensemble_learning(model, $M_{3d}$);
return $E$;
```

Algorithm 2 Pseudo-code for the majority voting algorithm.

```
Input: $C_Q$ in (7)
Output: majority class $C_M$
Set the class set $S = [1, 2, \cdots, N_C]$;
# Here, elements in $S$ indicate BMI classes.
Create a 1-by-$N_C$ vector of zeros $V = [0, 0, \cdots, 0]$;
Initialize the majority class $C_M$ to 0;
# Count the frequency of each class in $C_Q$.
for $q$ = 1:1:size($C_Q$) do
    $V[C_Q[q]] = V[C_Q[q]] + 1$;
end for
Set $C_M$ to the index that corresponds to the maximum value in $V$;
return $C_M$;
```

5. Experimental Setup

We evaluated the proposed method on an existing publicly available BMI classification dataset [28]. This dataset is a unique BMI classification dataset that contains 3D human skeleton sequences as well as the body weights and heights of subjects. To assess the performance of the proposed ensemble model, we used five-fold cross-validation and leave-one-person-out cross-validation.

5.1. Five-Fold Cross Validation

The dataset consists of 112 people with five skeleton sequences for each person. However, for one individual, denoted as "Person158", only four sequences exist. To implement five-fold cross-validation, we decided to eliminate this individual from the dataset. Additionally, there are six sequences for the following four individuals: “Person034”, “Person036”, “Person053” and “Person096”. After examining the six sequences for each of these individuals, we discarded the noisiest sequence for each individual. As a result, the first sequence for “Person034”, the third sequence for “Person036”, the fifth sequence for “Person053” and the sixth sequence for “Person096” were excluded from the dataset. As a result, the final dataset contained a total of 555 sequences (equaling 111 people × 5 sequences per person).

By using the body weights and heights of the 111 people mentioned above, we calculated the BMI for each individual and categorized the people according to their nutritional statuses. As shown in Table 1, only one person was categorized as “obesity class II”, and there were no people categorized as “obesity class III”. We decided to group “pre-obesity”, “obesity class I”, “obesity class II” and “obesity class III” into one class, called “overweight”. As a result, the number of people included in the overweight class was $32(= 25 + 6 + 1)$. In our experiments, the people included in the class of “underweight” were labeled as “1”. The people included in the classes of “normal weight” and overweight were labeled as “2” and “3”, respectively.

Table 1. Number of people for each nutritional status.

BMI	Nutritional Status		# of Persons
Below 18.5	Underweight		6
18.5–24.9	Normal weight		73
25.0–29.9	Obesity	Pre-obesity	25
30.0–34.9		Obesity class I	6
35.0–39.9		Obesity class II	1
Above 40.0		Obesity class III	0

In each cross-validation fold, for each person, four sequences were selected from the five available sequences. The selected 444 sequences (i.e., 111 people × 4 sequences per person) were used to train the proposed model. The remaining 111 (i.e., $555 - 444$) sequences were used to test the model.

Although the number of subjects in class 3 (overweight) was increased by consolidation, the three classes were still imbalanced. As shown in Table 1, the number of subjects in class 1 (underweight) was too low. The number of subjects in class 2 (normal weight) was greater than the numbers of subjects in the other classes. During the training phase of the proposed model, we used the SMOTE algorithm proposed in [41] to balance the three classes. For class 2, the number of anthropometric feature vectors was 292 (i.e., 73 people × 4 sequences per person). Therefore, according to the application of SMOTE, the numbers of anthropometric feature vectors for classes 1 and 3 were also 292. As a result, in each cross-validation fold, the model was trained using 876 anthropometric feature vectors (i.e., 292 vectors per class × 3 classes).

5.2. Leave-One-Person-Out Cross-Validation

Unlike five-fold cross-validation, all the sequences for 112 people were used in leave-one-person-out cross-validation. As a result, the dataset contained a total of 563 sequences (i.e., 107 people × 5 sequences per person + 1 person × 4 sequences per person + 4 people × 6 sequences per person). In addition, since the BMI of Person158 was greater than 25, Person158 was categorized as class 3 (overweight). As a result, the number of subjects included in classes 1, 2 and 3 were 6, 73 and 33, respectively. In each validation round, the skeleton sequences for one person were used to test the model, while the sequences for the remaining 111 people were used for training. In addition, since the three classes were imbalanced, we used the SMOTE algorithm to balance the three classes during the training phase of the proposed model.

5.3. Performance Measurement

In each cross-validation fold (round), to evaluate the performance of the proposed ensemble model, we used the following measures: the true positive rate (TPR), positive predictive value (PPV) and F_1 score. Here, for each class, the TPR, which is also called recall, is defined as

$$\mathrm{TPR} = \frac{\sum \text{True positive}}{\sum \text{Condition positive}}. \tag{8}$$

For each class, PPV, which is also called precision, is defined as

$$\mathrm{PPV} = \frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}. \tag{9}$$

Based on (8) and (9), the F_1 score for each class is calculated as

$$F_1 = 2 \cdot \frac{\mathrm{TPR} \cdot \mathrm{PPV}}{\mathrm{TPR} + \mathrm{PPV}}. \tag{10}$$

Additionally, the accuracy becomes

$$\text{Accuracy} = \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}. \tag{11}$$

Furthermore, we computed the macro-average values for TPR, PPV and F_1 for each cross-validation fold as follows:

$$\text{Macro-average of TPR} = \frac{\sum_{b=1}^{B} \mathrm{TPR}(\text{class } b)}{B}, \tag{12}$$

$$\text{Macro-average of PPV} = \frac{\sum_{b=1}^{B} \mathrm{PPV}(\text{class } b)}{B}, \tag{13}$$

$$\text{Macro-average of } F_1 = \frac{\sum_{b=1}^{B} F_1(\text{class } b)}{B}, \tag{14}$$

where B is the number of classes. Here, $B = 3$ because there are three classes: underweight (class 1), normal weight (class 2) and overweight (class 3).

6. Results

We investigated the main factors for the low accuracy of BMI classification reported in [28]. The main reason for this low accuracy was that Andersson et al. in [28] used an undersampling technique to overcome the class imbalance problem in their dataset. For benchmarking, we tested five machine learning algorithms, namely C-SVM, nu-SVM, k-NN, Naive Bayes (NB) and decision tree (DT), in conjunction with the undersampling technique. C-SVM and nu-SVM were implemented using LIBSVM (version 3.21), which is an open-source library for SVMs [51]. Additionally, according to the recommendations in [52], the radial basis function (RBF) kernel was used for the C-SVM and nu-SVM models. C-SVM has a cost parameter denoted as c whose value ranges from zero to infinity. nu-SVM has a regularization parameter denoted as g whose value lies within $[0, 1]$. The RBF kernel has a gamma parameter denoted as γ. The grid search method was used to find the best parameter combination for the C-SVM and nu-SVM models. The remaining k-NN, NB and DT models were implemented using the MATLAB functions “fitcknn”, “fitcnb” and “fitctree”, respectively. Additionally, to determine the best hyperparameter configuration for each model, we executed the hyperparameter optimization processes supported by the aforementioned functions. To this end, we used “Statistics and Machine Learning Toolbox”. In addition, the models were implemented and tested in MATLAB R2018a (9.4.0.813654).

6.1. Results of Five-Fold Cross-Validation

During the development of the proposed ensemble model, determining the optimal number of k-NN models to be used was a significant challenge. To this end, we constructed several ensemble models with different numbers of k-NN models and compared the resulting performance metrics. To implement each ensemble model, we used the MATLAB function "fitcknn". To this end, we used "Statistics and Machine Learning Toolbox". To determine the optimal hyperparameter configuration for each k-NN model, we performed hyperparameter optimization using the aforementioned function. For the hyperparameter optimization process, there were five optimizable parameters: the value of k, the distance metric, the distance weighting function, the Minkowski distance exponent and a flag to standardize predictors. During the training phase for each model, these five parameters were optimized using the hyperparameter optimization process. The proposed ensemble model was implemented and tested in MATLAB R2018a (9.4.0.813654). The MATLAB code is available at https://sites.google.com/view/beomkwon/bmi-classification.

Table 2 lists the performance metrics for the ensemble models with different numbers of k-NN models. A detailed description of the experimental setup is listed in Table 3. In Table 2, the proposed ensemble model consists of 15 k-NN$_q$ models ($q \in \{1, 2, \cdots, 14, 15\}$). "Comparison model #1" consists of three k-NN$_q$ models ($q \in \{1, 2, 3\}$). "Comparison model #2" consists of six k-NN$_q$ models ($q \in \{1, 2, \cdots, 5, 6\}$). "Comparison model #3" consists of 10 k-NN$_q$ models ($q \in \{1, 2, \cdots, 9, 10\}$). "Comparison model #4" consists of 16 k-NN$_q$ models ($q \in \{1, 2, \cdots, 15, 16\}$). "Comparison model #5" consists of 17 k-NN$_q$ models ($q \in \{1, 2, \cdots, 16, 17\}$). "Comparison model #6" consists of 18 k-NN$_q$ models ($q \in \{1, 2, \cdots, 17, 18\}$). "Comparison model #7" consists of 19 k-NN$_q$ models ($q \in \{1, 2, \cdots, 18, 19\}$). "Comparison model #8" consists of 20 k-NN$_q$ models ($q \in \{1, 2, \cdots, 19, 20\}$). "Comparison model #9" consists of 21 k-NN$_q$ models ($q \in \{1, 2, \cdots, 20, 21\}$). One can see that the performance of the ensemble model increased as the number of k-NN models increased. However, when the number of k-NN models was greater than 15, the performance of the ensemble model was not improved. Based on these results, we selected an ensemble model consisting of 15 k-NN models for additional testing. However, as shown in Table 2, since the average running time of the ensemble model increased as the number of k-NN models increased, the trade-off between classification accuracy and running time needs to be considered in real-world applications.

Table 2. Five-fold cross-validation performance comparisons between ensemble models with different numbers of k-NN models. TPR: true positive rate; PPV: positive predictive value.

Method	# of k-NNs	Macro-Average			Accuracy	Running Time
		TPR	PPV	F_1		
Comparison model #1	3	0.9588	0.8676	0.9001	0.9441	0.5565 s
Comparison model #2	6	0.9677	0.9092	0.9315	0.9640	0.9856 s
Comparison model #3	10	0.9790	0.9428	0.9585	0.9748	1.5512 s
Proposed ensemble model	15	0.9862	0.9603	0.9721	0.9820	2.2610 s
Comparison model #4	16	0.9758	0.9566	0.9646	0.9730	2.3994 s
Comparison model #5	17	0.9746	0.9575	0.9643	0.9730	2.5386 s
Comparison model #6	18	0.9776	0.9595	0.9669	0.9766	2.6790 s
Comparison model #7	19	0.9746	0.9575	0.9643	0.9730	2.8217 s
Comparison model #8	20	0.9686	0.9585	0.9627	0.9766	2.9629 s
Comparions model #9	21	0.9809	0.9594	0.9689	0.9784	3.1046 s

Table 3. Experimental setup.

Item	Explanation
Operating System	Windows 10 Pro
Processor	Inter(R) Core(TM) i5-8250U CPU @ 1.60 GHz
Memory	8.00 GB
GPU	Intel(R) UHD Graphics 620

Table 4 lists the optimized parameter settings for the ensemble model consisting of 15 k-NN models in each cross-validation fold.

Table 4. Parameter settings for the proposed ensemble model in five-fold cross-validation.

	Fold #1	Fold #2	Fold #3	Fold #4	Fold #5
k-NN$_1$	(1, m, in, 0.7140, 0)	(1, c_3, sq, -, 1)	(1, s, sq, -, 0)	(1, c_3, sq, -, 1)	(1, c_1, sq, -, 0)
k-NN$_2$	(1, m, eq, 0.7075, 1)	(1, c_2, sq, -, 1)	(1, c_3, sq, -, 1)	(1, c_3, eq, -, 1)	(1, e, sq, -, 1)
k-NN$_3$	(2, c_3, in, -, 1)	(1, c_2, sq, -, 1)	(1, m, sq, 0.6965, 1)	(1, e, sq, -, 1)	(1, e, in, -, 1)
k-NN$_4$	(1, c_3, in, -, 1)	(1, m, eq, 0.5485, 1)	(1, c_3, in, -, 1)	(1, c_3, sq, -, 1)	(1, m, sq, 0.6046, 0)
k-NN$_5$	(1, c_3, sq, -, 1)	(1, c_3, sq, -, 1)	(2, c_3, in, -, 1)	(1, m, sq, 1.2213, 1)	(3, c_3, sq, -, 1)
k-NN$_6$	(1, m, sq, 0.7156, 1)	(1, c_3, in, -, 1)	(1, m, eq, 0.7499, 1)	(1, m, sq, 1.1226, 1)	(1, c_1, sq, -, 1)
k-NN$_7$	(1, m, sq, 0.7317, 0)	(1, s, in, -, 0)	(1, e, sq, -, 1)	(1, s, sq, -, 0)	(1, m, eq, 0.8746, 1)
k-NN$_8$	(1, e, sq, -, 1)	(1, c_3, sq, -, 1)	(1, m, sq, 0.7044, 1)	(1, c_1, sq, -, 1)	(1, m, in, 0.5390, 1)
k-NN$_9$	(1, e, sq, -, 1)	(1, c_1, sq, -, 1)	(1, e, sq, -, 1)	(1, m, eq, 1.5315, 1)	(1, c_2, in, -, 1)
k-NN$_{10}$	(1, m, in, 0.9299, 0)	(1, c_3, in, -, 1)	(1, c_3, sq, -, 1)	(1, m, in, 0.6912, 0)	(1, m, eq, 1.3160, 0)
k-NN$_{11}$	(1, m, in, 0.7098, 1)	(1, m, eq, 1.7090, 1)	(1, c_2, sq, -, 1)	(1, m, eq, 0.8312, 0)	(1, s, sq, -, 0)
k-NN$_{12}$	(1, e, sq, -, 1)	(1, m, sq, 1.0923, 1)	(1, e, sq, -, 1)	(1, m, sq, 1.1597, 1)	(1, e, sq, -, 1)
k-NN$_{13}$	(1, c_3, sq, -, 1)	(1, m, in, 1.4446, 0)	(2, c_3, in, -, 1)	(1, e, sq, -, 1)	(2, c_1, in, -, 0)
k-NN$_{14}$	(1, c_1, eq, -, 0)	(1, e, sq, -, 1)	(1, m, sq, 1.0183, 1)	(1, c_1, sq, -, 1)	(1, m, sq, 1.3201, 1)
k-NN$_{15}$	(1, e, sq, -, 1)	(1, s, eq, -, 0)	(1, c_3, sq, -, 1)	(1, s, sq, -, 0)	(1, c_3, sq, -, 1)

Note: Elements in brackets indicate k, distance metric, distance weighting function, Minkowski distance exponent, and flag to standardize predictors, respectively. In this table, the distance metrics "cityblock", "correlation", "cosine", "euclidean", "minkowski" and "seuclidean" are represented as "c_1", "c_2", "c_3", "e", "m", and "s", respectively. The distance weighting functions "equal", "inverse" and "squaredinverse" are denoted as "eq", "in" and "sq", respectively. k-NN: k-nearest neighbor.

Table 5 lists the five-fold cross-validation results of the five benchmark methods discussed above. Here, instead of SMOTE, the undersampling technique was used for performance evaluations. Therefore, during the training phase of each cross-validation fold, for class 2, 268 (i.e, 73 × 4 − 6 × 4) anthropometric feature vectors were randomly selected among the 292 total vectors and removed from the training dataset. For class 3, 104 (i.e., 32 × 4 − 6 × 4) vectors were randomly selected among the 128 total vectors and removed from the training dataset. The remaining 72 (i.e., 24 anthropometric feature vectors per class × 3 classes) vectors were used to train the algorithm for each method. As shown in the table, for each method, there were significant variations in the TPR, PPV and F_1 values over the three classes. These results demonstrate that the undersampling technique used in [28] is not effective in overcoming the class imbalance problem.

Table 5. Five-fold cross-validation results of the five benchmark methods when the undersampling technique is used. SVM: support vector machine; NB: naive Bayes; DT: decision tree.

Method	Class	Fold #1			Fold #2			Fold #3			Fold #4			Fold #5		
		TPR	PPV	F_1	TPR	PPV	F_1	TPR	PPV	F_1	TPR	PPV	F_1	TPR	PPV	F_1
C-SVM	1	1.00	0.17	0.29	0.67	0.80	0.73	0.67	0.67	0.67	0.67	1.00	0.80	0.83	0.71	0.77
	2	0.41	0.86	0.56	0.92	0.77	0.84	0.86	0.79	0.82	0.92	0.79	0.85	0.95	0.79	0.86
	3	0.56	0.44	0.49	0.41	0.68	0.51	0.53	0.68	0.60	0.50	0.73	0.59	0.41	0.76	0.53
nu-SVM	1	0.67	0.36	0.47	0.67	0.57	0.62	0.67	0.67	0.67	0.67	1.00	0.80	0.83	0.71	0.77
	2	0.75	0.80	0.77	0.86	0.78	0.82	0.84	0.80	0.82	0.96	0.76	0.85	0.93	0.80	0.86
	3	0.53	0.55	0.54	0.44	0.61	0.51	0.59	0.66	0.62	0.38	0.80	0.51	0.44	0.74	0.55
k-NN [28]	1	1.00	0.27	0.43	1.00	0.25	0.40	0.83	0.17	0.28	1.00	0.23	0.38	1.00	0.25	0.40
	2	0.45	0.77	0.57	0.52	0.90	0.66	0.47	0.89	0.61	0.41	0.75	0.53	0.52	0.83	0.64
	3	0.59	0.41	0.49	0.75	0.53	0.62	0.66	0.49	0.56	0.56	0.40	0.47	0.59	0.46	0.52
NB	1	0.33	0.10	0.15	0.33	0.08	0.13	0.50	0.11	0.18	0.83	0.08	0.14	0.83	0.37	0.59
	2	0.41	0.81	0.55	0.12	0.56	0.20	0.52	0.75	0.61	0.29	0.81	0.42	0.19	0.77	0.38
	3	0.63	0.38	0.47	0.69	0.32	0.44	0.41	0.41	0.41	0.28	0.43	0.34	0.31	0.50	0.46
DT	1	0.33	0.06	0.10	0.67	0.14	0.23	0.50	0.10	0.17	0.50	0.17	0.25	0.83	0.13	0.23
	2	0.47	0.71	0.56	0.38	0.82	0.52	0.44	0.74	0.55	0.42	0.63	0.51	0.29	0.64	0.40
	3	0.38	0.41	0.39	0.53	0.35	0.43	0.38	0.32	0.34	0.44	0.32	0.37	0.47	0.38	0.42

Table 6 lists the five-fold cross-validation results of the five benchmark methods when SMOTE is used to alleviate the class imbalance problem. The results in this table demonstrate that SMOTE improves the TPR, PPV and F_1 score values of the five methods compared to the results in Table 5. By using (12) to (14), we computed the average metric values over five-fold cross-validation, as shown in Table 7. The results in this table demonstrate that it is better to use SMOTE instead of the undersampling technique to alleviate the class imbalance problem. For all benchmark methods, average performance improvements are achieved when SMOTE is applied. In particular, the *k*-NN model outperforms the other methods, achieving results of TPR = 0.9276, PPV = 0.8512, F_1 = 0.8798 and accuracy = 0.9279. Based on these results, we decided to use *k*-NN models to construct the proposed ensemble model.

Table 6. Five-fold cross-validation results of the five benchmark methods when the synthetic minority oversampling technique (SMOTE) is used.

Method	**Class**	**Fold #1**			**Fold #2**			**Fold #3**			**Fold #4**			**Fold #5**		
		TPR	**PPV**	F_1	**TPR**	**PPV**	F_1	**TPR**	**PPV**	F_1	**TPR**	**PPV**	F_1	**TPR**	**PPV**	F_1
C-SVM	1	0.50	0.75	0.60	0.67	1.00	0.80	0.67	1.00	0.80	0.83	1.00	0.91	1.00	0.75	0.86
	2	0.97	0.85	0.90	0.96	0.93	0.95	1.00	0.95	0.97	0.97	0.99	0.98	0.99	0.92	0.95
	3	0.66	0.91	0.76	0.91	0.91	0.91	0.91	0.97	0.94	1.00	0.94	0.97	0.75	0.96	0.84
nu-SVM	1	0.50	0.75	0.60	0.67	1.00	0.80	0.67	1.00	0.80	0.83	1.00	0.91	1.00	0.75	0.86
	2	0.96	0.85	0.90	0.95	0.92	0.93	1.00	0.95	0.97	0.97	0.99	0.98	0.99	0.94	0.96
	3	0.69	0.88	0.77	0.88	0.88	0.88	0.91	0.97	0.94	1.00	0.94	0.97	0.78	0.96	0.86
k-NN [28]	1	0.83	0.50	0.63	1.00	0.86	0.92	0.83	0.71	0.77	1.00	0.75	0.86	1.00	0.55	0.76
	2	0.90	0.93	0.92	0.97	0.97	0.97	0.96	1.00	0.98	0.93	1.00	0.96	0.92	0.97	0.94
	3	0.78	0.83	0.81	0.91	0.94	0.92	1.00	0.94	0.97	1.00	0.91	0.96	0.88	0.90	0.89
NB	1	0.33	0.13	0.19	0.17	0.13	0.14	0.50	0.38	0.43	0.33	0.29	0.31	0.67	0.27	0.38
	2	0.68	0.79	0.74	0.78	0.79	0.79	0.78	0.83	0.80	0.73	0.87	0.79	0.64	0.87	0.74
	3	0.53	0.52	0.52	0.63	0.65	0.63	0.63	0.59	0.61	0.78	0.58	0.67	0.66	0.50	0.57
DT	1	0.17	0.10	0.13	0.50	0.20	0.29	0.50	0.43	0.46	0.50	0.43	0.46	0.67	0.31	0.42
	2	0.71	0.71	0.71	0.73	0.79	0.76	0.77	0.77	0.77	0.82	0.90	0.86	0.70	0.78	0.74
	3	0.38	0.43	0.40	0.53	0.59	0.56	0.47	0.48	0.48	0.75	0.65	0.70	0.53	0.52	0.52

Table 7. Average performance comparisons of five benchmark methods over five-fold cross validation.

Method	**Macro-Average**			**Accuracy**
	TPR	**PPV**	F_1	
C-SVM + undersampling	0.6863	0.7096	0.6604	0.7135
C-SVM + SMOTE	0.8517	0.9215	0.8760	0.9261
nu-SVM + undersampling	0.6812	0.7069	0.6783	0.7459
nu-SVM + SMOTE	0.8520	0.9178	0.8752	0.9243
k-NN + undersampling [28]	0.6906	0.5074	0.5035	0.5459
k-NN + SMOTE	0.9276	0.8512	0.8798	0.9297
NB + undersampling	0.4760	0.4106	0.3537	0.4054
NB + SMOTE	0.5890	0.5444	0.5536	0.6829
DT + undersampling	0.4681	0.3945	0.3640	0.4198
DT + SMOTE	0.5810	0.5385	0.5493	0.6685

Figure 6 presents the confusion matrices for the proposed ensemble model for each cross-validation fold. The diagonal elements in each confusion matrix indicate the numbers of correctly classified samples. The other elements indicate the numbers of incorrectly classified samples. One can see that the proposed ensemble model classifies class 1 as the minority class among the three classes without misclassification. Additionally, for class 3 (the second minority class), the ensemble model also exhibits a low misclassification rate.

		True class			
	class	1	2	3	Total
Predicted class	1	6	1	0	7
	2	0	70	4	74
	3	0	2	28	30
	Total	6	73	32	111

(**a**) Fold #1

		True class			
	class	1	2	3	Total
Predicted class	1	6	0	1	7
	2	0	72	0	72
	3	0	1	31	32
	Total	6	73	32	111

(**b**) Fold #2

		True class			
	class	1	2	3	Total
Predicted class	1	6	0	0	6
	2	0	73	0	73
	3	0	0	32	32
	Total	6	73	32	111

(**c**) Fold #3

		True class			
	class	1	2	3	Total
Predicted class	1	6	0	0	6
	2	0	73	0	73
	3	0	0	32	32
	Total	6	73	32	111

(**d**) Fold #4

		True class			
	class	1	2	3	Total
Predicted class	1	6	0	0	6
	2	0	73	1	74
	3	0	0	31	31
	Total	6	73	32	111

(**e**) Fold #5

Figure 6. Confusion matrices for the proposed ensemble model for different cross-validation folds.

Based on the results in the confusion matrices in Figure 6, we computed the TPR, PPV and F_1 score values of each class for five-fold cross-validation. Additionally, we computed the macro-average values of these metrics and classification accuracy. The performance evaluation results for the proposed ensemble model are summarized in Table 8. As shown in this table, the proposed model exhibits a robust and accurate classification performance for the minority class and majority class, achieving approximately 98.2% average classification accuracy.

Table 8. Results of five-fold cross-validation for the proposed ensemble model.

Fold #	**Class**	**TPR**	**PPV**	F_1	**Macro-Average**			**Accuracy**
					TPR	**PPV**	F_1	
1	Underweight Normal weight Overweight	1.0000 0.9452 0.9063	0.8571 0.9583 0.9063	0.9231 0.9517 0.9063	0.9505	0.9072	0.9270	0.9369
2	Underweight Normal weight Overweight	1.0000 0.9863 0.9688	0.8571 1.0000 0.9688	0.9231 0.9931 0.9688	0.9850	0.9420	0.9616	0.9820
3	Underweight Normal weight Overweight	1.0000 1.0000 1.0000	1.0000 1.0000 1.0000	1.0000 1.0000 1.0000	1.0000	1.0000	1.0000	1.0000
4	Underweight Normal weight Overweight	1.0000 1.0000 1.0000	1.0000 1.0000 1.0000	1.0000 1.0000 1.0000	1.0000	1.0000	1.0000	1.0000
5	Underweight Normal weight Overweight	1.0000 0.9863 1.0000	0.8571 1.0000 1.0000	0.9231 0.9931 1.0000	0.9954	0.9524	0.9721	0.9910
				Average	0.9862	0.9603	0.9721	0.9820

Table 9 lists the macro-average values of TPR, PPV and F_1 scores, as well as the accuracy of each method over five-fold cross-validation. One can see that the proposed ensemble model performs best in terms of all evaluation metrics. This is because the ensemble model is trained using anthropometric features calculated over long/mid/short-term periods. In other words, the use of such features enables the ensemble model to be trained effectively by minimizing the adverse effects of variance in extracted features. As a result, the proposed model can achieve robust and accurate BMI classification performance. Among the considered benchmark methods, the *k*-NN model achieves the best performance, whereas DT achieves the worst performance. The classification accuracy of the proposed model is approximately 5.23% greater than that of a single *k*-NN model.

Table 9. Five-fold cross-validation performance comparisons between the proposed ensemble model and five benchmark methods.

Method	Macro-Average			Accuracy
	TPR	PPV	F_1	
C-SVM	0.8517	0.9215	0.8760	0.9261
nu-SVM	0.8520	0.9178	0.8752	0.9243
k-NN [28]	0.9276	0.8512	0.8798	0.9297
NB	0.5890	0.5444	0.5536	0.6829
DT	0.5810	0.5385	0.5493	0.6685
Proposed	0.9862	0.9603	0.9721	0.9820

To verify the benefits of using the anthropometric features calculated for various different periods in the BMI classification task, we analyzed the standard deviations of the anthropometric features. For explanation, let $AF(r)$ be AF for the rth skeleton sequence in the dataset. Here, according to the definition of AF in (6), the dimension of $AF(r)$ is 20. In addition, let $AF(r)[u]$, $u \in \{1, 2, \cdots, 19, 20\}$, be the uth element of $AF(r)$. Then, the average value of $AF(r)[u]$ over the whole sequences can be obtained as

$$AVG[u] = \sum_{r=1}^{R} AF(r)[u], \;\; u \in \{1, 2, \cdots, 19, 20\}, \tag{15}$$

where R is the total number of skeleton sequences.

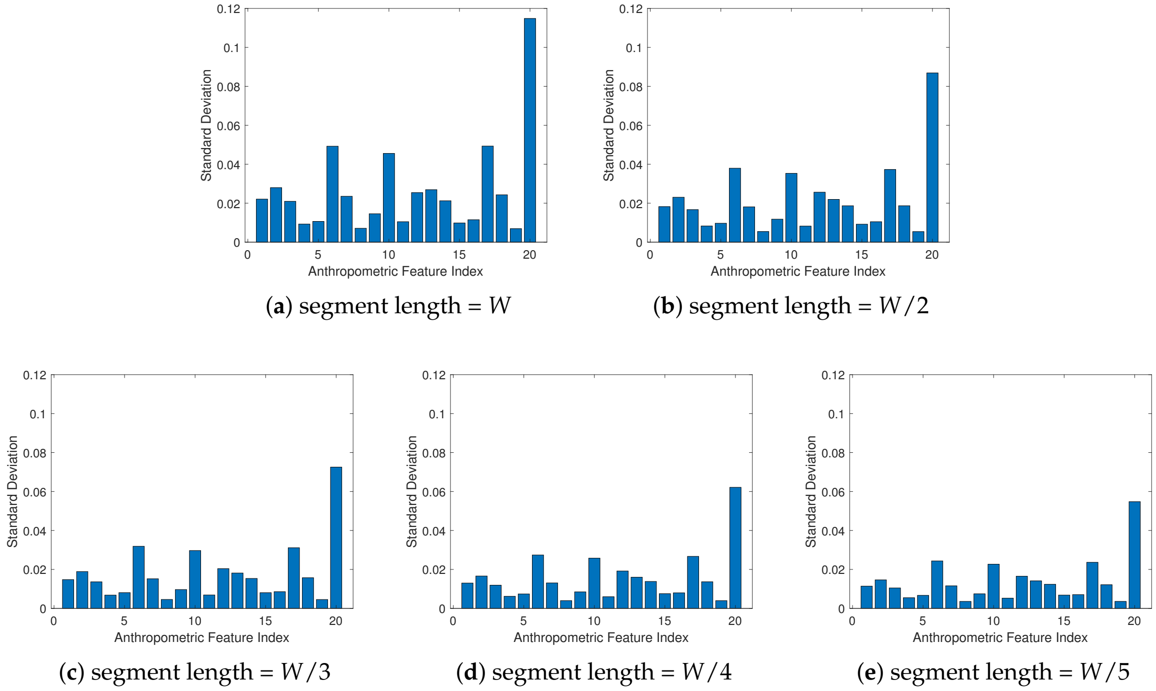
Based on (15), the standard deviation of each of the 20 anthropometric features is calculated as

$$STD[u] = \sqrt{\frac{\sum_{r=1}^{R}\{AF(r)[u] - AVG[u]\}^2}{R-1}}, \;\; u \in \{1, 2, \cdots, 19, 20\}, \tag{16}$$

Figure 7 presents the standard deviations of the 20 anthropometric features for five different periods (i.e., W, $W/2$, $W/3$, $W/4$ and $W/5$). In this figure, for the cases of $W/2$, $W/3$, $W/4$ and $W/5$, we calculated the average of the standard deviations. For example, for $W/5$, there are five equal-length segments according to the frame division process of the proposed method. We calculated the standard deviations in (16) for each of the segements, and then calculated the average of them over the five segments. As shown in Figure 7a, the features have high standard deviations when they are calculated for all frames (i.e., W). In contrast, the features calculated for $W/2$, $W/3$, $W/4$ and $W/5$ have relatively low standard deviations. In particular, the features calculated for $W/5$ have the lowest standard deviation values. A low standard deviation indicates that the features are clustered around the average. A high standard deviation means that the features are spread out over a wide range. Because the anthropometric features are calculated as average lengths over a given segment sequence, the high standard deviations shown in Figure 7a may adversely affect the performance of machine learning algorithms. In contrast, in our ensemble learning method, anthropometric features with low standard deviations are used to train/test multiple *k*-NN models. Based on the use of these features, the proposed ensemble model can be trained effectively by minimizing the adverse effects of variance in extracted features, resulting in state-of-the-art performance.

6.2. Results of Leave-One-Person-Out Cross-Validation

Figure 8 shows the leave-one-person-out cross-validation process used in this work. As shown in the figure, in each validation round, the skeleton sequences for one person were used to test the classifier, and the sequences for the remaining 111 people were used for training. In each validation round, the predicted results (i.e., predicted BMI classes) for the testing skeleton sequences were obtained. After all 112 validations rounds were completed, we calculated a confusion matrix using all predicted results. Based on the confusion matrix, we calculated the TPR, PPV and F_1 values over the three classes in order to evaluate the performance of the classifier.



(**a**) segment length = W (**b**) segment length = $W/2$

(**c**) segment length = $W/3$ (**d**) segment length = $W/4$ (**e**) segment length = $W/5$

Figure 7. Standard deviations of 20 anthropometric features for long/mid/short-term periods.

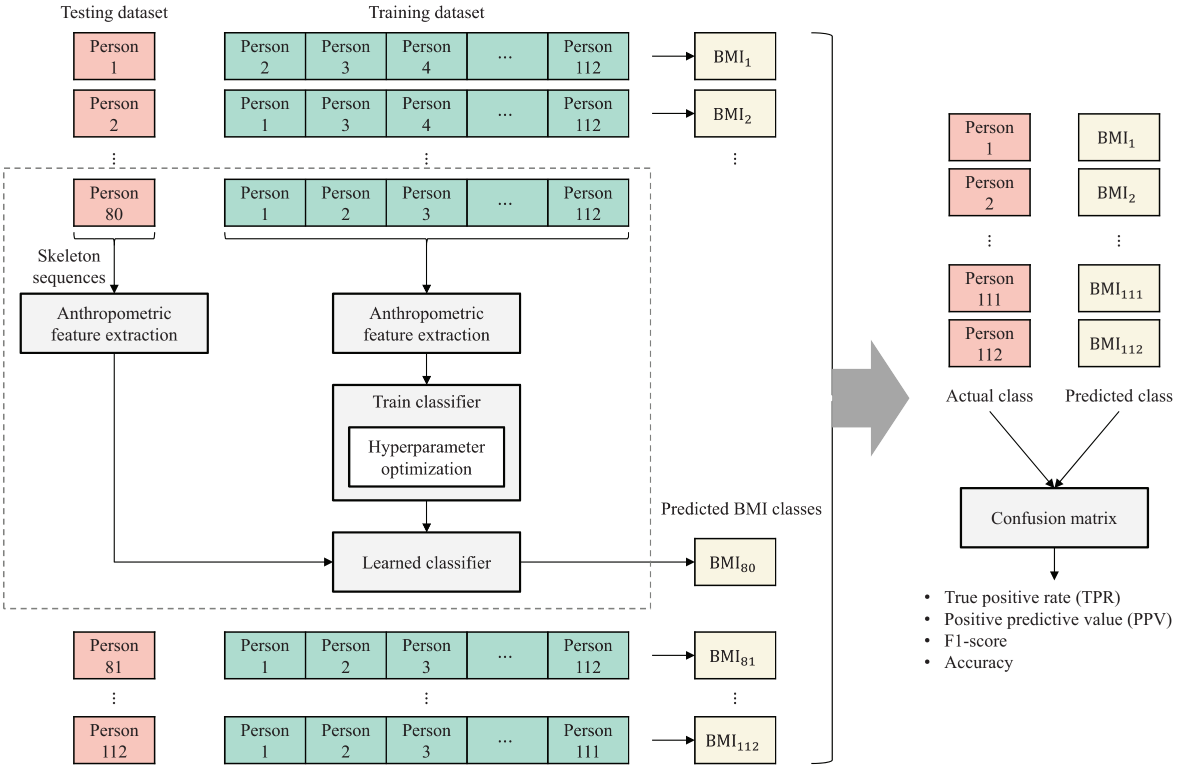


Figure 8. Leave-one-person-out cross-validation process to evaluate the performance of each method.

Table 10 shows the leave-one-person-out cross-validation results of the five benchmark methods when the undersampling technique was used. On the other hand, Table 11 shows the leave-one-person-out cross-validation results when SMOTE was used. By using (12) to (14), we computed the macro-average values of these metrics over the three classes. In addition, we calculated the classification accuracy of each benchmark method. The performance evaluation results for the five benchmark methods are summarized in Table 12. From the table, it is seen that SMOTE improves the macro-average of the TPR, PPV and F_1 of each benchmark method, compared with the results where the undersampling technique was used. In addition, the BMI classification accuracy of each method also improved when SMOTE was used. These results demonstrate that SMOTE is more effective at overcoming the class imbalance problem in the dataset used in [28] than the undersampling technique.

Table 10. Leave-one-person-out cross-validation results of the five benchmark methods when the undersampling technique is used.

Method	Class	TPR	PPV	F_1	Macro-Average			Accuracy
					TPR	PPV	F_1	
C-SVM	Underweight	0.2667	0.0563	0.0930	0.3397	0.3493	0.3057	0.3837
	Normal weight	0.4130	0.7273	0.5269				
	Overweight	0.3394	0.2642	0.2971				
nu-SVM	Underweight	0.2667	0.0533	0.0889	0.3361	0.3474	0.3019	0.3766
	Normal weight	0.4022	0.7184	0.5157				
	Overweight	0.3394	0.2705	0.3011				
k-NN [28]	Underweight	0.2000	0.0405	0.0674	0.3054	0.3162	0.2774	0.3410
	Normal weight	0.3342	0.6150	0.4331				
	Overweight	0.3818	0.2930	0.3316				
NB	Underweight	0.1000	0.0211	0.0349	0.2616	0.2949	0.2606	0.3393
	Normal weight	0.3696	0.5913	0.4548				
	Overweight	0.3152	0.2723	0.2921				
DT	Underweight	0.2000	0.0405	0.0674	0.3054	0.3162	0.2774	0.3410
	Normal weight	0.3342	0.6150	0.4331				
	Overweight	0.3818	0.2930	0.3316				

Table 11. Leave-one-person-out cross-validation results of the five benchmark methods when SMOTE is used.

Method	Class	TPR	PPV	F_1	Macro-Average			Accuracy
					TPR	PPV	F_1	
C-SVM	Underweight	0.1667	0.1163	0.1370	0.3981	0.3904	0.3926	0.5560
	Normal weight	0.6821	0.6896	0.6858				
	Overweight	0.3455	0.3654	0.3551				
nu-SVM	Underweight	0.0333	0.0303	0.0317	0.3620	0.3662	0.3633	0.5719
	Normal weight	0.7255	0.6881	0.7063				
	Overweight	0.3273	0.3803	0.3518				
k-NN [28]	Underweight	0.3000	0.2000	0.2400	0.4852	0.4536	0.4627	0.5790
	Normal weight	0.6223	0.7316	0.6725				
	Overweight	0.5333	0.4293	0.4757				
NB	Underweight	0.0333	0.0227	0.0270	0.3404	0.3450	0.3422	0.5187
	Normal weight	0.6304	0.6610	0.6453				
	Overweight	0.3576	0.3512	0.3544				
DT	Underweight	0.0769	0.1000	0.0870	0.3387	0.3411	0.3394	0.5062
	Normal weight	0.6474	0.6386	0.6430				
	Overweight	0.2919	0.2848	0.2883				

Table 12. Performance comparisons of the five benchmark methods over leave-one-person-out cross validation.

Method	Macro-Average			Accuracy
	TPR	PPV	F_1	
C-SVM + undersampling	0.3397	0.3493	0.3057	0.3837
C-SVM + SMOTE	0.3981	0.3904	0.3926	0.5560
nu-SVM + undersampling	0.3361	0.3474	0.3019	0.3766
nu-SVM + SMOTE	0.3620	0.3662	0.3633	0.5719
k-NN + undersampling [28]	0.3054	0.3162	0.2774	0.3410
k-NN + SMOTE	0.4852	0.4536	0.4627	0.5790
NB + undersampling	0.2616	0.2949	0.2606	0.3393
NB + SMOTE	0.3404	0.3450	0.3422	0.5187
DT + undersampling	0.3054	0.3162	0.2774	0.3410
DT + SMOTE	0.3387	0.3411	0.3394	0.5062

To find the optimal number of *k*-NN classifiers in the proposed ensemble model, we evaluated the performance of the ensemble models with different numbers of *k*-NN classifiers. Table 13 lists the performance metrics for each model. From this table, it can be seen that the best performance was achieved when the ensemble model consisted of 15 k-NN$_q$ models ($q \in \{1, 2, \cdots, 14, 15\}$). Based on these results, we selected an ensemble model consisting of 15 *k*-NN models for additional testing.

Table 13. Leave-one-person-out cross-validation performance comparisons between ensemble models with different numbers of *k*-NN classifiers.

Method	# of *k*-NNs	Macro-Average			Accuracy
		TPR	PPV	F_1	
Comparison model #1	3	0.4889	0.4559	0.4646	0.5950
Comparison model #2	6	0.5300	0.4997	0.5030	0.6448
Comparison model #3	10	0.5825	0.5450	0.5579	0.6927
Proposed ensemble model	15	0.6422	0.5878	0.6058	0.7300
Comparions model #4	21	0.5864	0.5564	0.5676	0.7069

Table 14 lists the macro-average values of TPR, PPV and F_1 scores, as well as the accuracy of each method over leave-one-person-out cross-validation. As shown in the table, for all evaluation metrics, the proposed method outperforms the other methods, achieving a BMI classification accuracy of 73%. In the proposed method, anthropometric features calculated over long/mid/short-term periods were used for the training of the ensemble model. In addition, the use of such features in the training phase could reduce the adverse effects of a variance in extracted features. As a result, the model could be trained effectively and achieved the best performance among the methods.

Table 14. Leave-one-person-out cross-validation performance comparisons between the proposed ensemble model and five benchmark methods.

Method	Macro-Average			Accuracy
	TPR	PPV	F_1	
C-SVM	0.3981	0.3904	0.3926	0.5560
nu-SVM	0.3620	0.3662	0.3633	0.5719
k-NN [28]	0.4852	0.4536	0.4627	0.5690
NB	0.3404	0.3450	0.3422	0.5187
DT	0.3387	0.3411	0.3394	0.5062
Proposed	0.6422	0.5878	0.6058	0.7300

7. Discussion

In this study, we used human body joints estimated from depth images captured by the Kinect (v1) sensor [42]. These estimated joints generally contain the noise and error introduced by an uncertainty. The majority of the uncertainty is caused by occlusion and depth ambiguity [17]. Due to this uncertainty, the values of anthropometric features vary by measurement time, even if they are extracted from the same parts of the human body. In conventional methods, for each anthropometric feature, the average value is calculated over the whole period. Then, the average values for the features are used as inputs to classifiers. The performance of classifiers trained using these values can be affected by the variance of features. In order to minimize this adverse effect, we propose the use of anthropometric features calculated over several different time periods. However, after the Kinect v1 sensor was released in 2010, the Kinect v2 sensor was released in 2014, and Wang et al. [53] evaluated the human body joint estimation accuracies of the Kinect v1 and v2 sensors; according to the results in [53], compared with the Kinect v1 sensor, the Kinect v2 sensor had better accuracy in joint estimation and showed more robust results for occlusion and body rotation. Recently, the new version of the Kinect sensor, called Azure Kinect, was released in 2019. Albert et al. [54] reviewed the improved hardware implementation and motion tracking algorithm of the Azure Kinect sensor. The authors also evaluated the motion tracking performances of the Kinect v2 and Azure Kinect sensors. Their results demonstrated that the Azure Kinect sensor achieved higher accuracy in human motion tracking than the Kinect v2 sensor. Furthermore, with the advancements in camera techniques, releases of new sensors with better hardware and motion tracking algorithms will continue. Due to the use of state-of-the-art sensors, if the noise and error in the estimated joints are reduced and the anthropometric features extracted at each time all become exactly equal, the feature vectors calculated over different time periods by the method will all become equal. Therefore, the usefulness and effectiveness of our proposed method could be limited in this case.

8. Conclusions

In general, there are variances in anthropometric features calculated over all frames of a sequence as a result of the inaccuracy in pose estimation. These variances adversely affect the classification performance of machine learning algorithms. Therefore, to minimize these adverse effects, we proposed the use of anthropometric features that are calculated over long/mid/short-term periods. Additionally, we proposed a novel ensemble model consisting of 15 *k*-NN models trained using features calculated over several different time periods. Experimental results demonstrated that the proposed ensemble model outperforms five benchmark methods and achieves state-of-the-art performance on a publicly available BMI classification dataset. In practical situations, some joints can be invisible due to occlusions. If the joints are invisible over the whole time interval, the corresponding anthropometric features cannot be obtained. In this case, the anthropometric feature vector may contain "not a number", called NaN. As the occlusions become severe, the number of such NaN values in the anthropometric feature vector increases. The use of these vectors in BMI classification can adversely affect the classification accuracy of the ensemble model. However, in this study, we considered only the condition that there was no occlusion and that all 20 joints were visible during the whole time interval. Therefore, as part of our future work, we plan to extend our method to classify the BMIs of people for whom some body parts are occluded.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, and writing—original draft preparation, B.K.; writing—review and editing, B.K. and S.L.; supervision and funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government (Ministry of Science and ICT, MSIT) under Grant NRF-2020R1A2C3011697.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Nguyen, N.T.; Nguyen, X.M.T.; Lane, J.; Wang, P. Relationship between obesity and diabetes in a US adult population: Findings from the national health and nutrition examination survey, 1999–2006. *Obes. Surg.* **2011**, *21*, 351–355. [PubMed]
2. Chen, Y.; Rennie, D.C.; Lockinger, L.A.; Dosman, J.A. Association between obesity and high blood pressure: Reporting bias related to gender and age. *Int. J. Obes.* **1998**, *22*, 771–777.
3. Lai, S.W.; Ng, K.C.; Lin, H.F.; Chen, H.L. Association between obesity and hyperlipidemia among children. *Yale J. Biol. Med.* **2001**, *74*, 205–210. [PubMed]
4. Thiet, M.D.; Mittelstaedt, C.A.; Herbst, C.A.; Buckwalter, J.A. Cholelithiasis in morbid obesity. *South. Med. J.* **1984**, *77*, 415–417. [PubMed]
5. De Sousa, A.G.P.; Cercato, C.; Mancini, M.C.; Halpern, A. Obesity and obstructive sleep apnea-hypopnea syndrome. *Obes. Rev.* **2008**, *9*, 340–354. [PubMed]
6. Magliano, M. Obesity and arthritis. *Menopause Int.* **2008**, *14*, 149–154.
7. Scott, K.M.; McGee, M.A.; Wells, J.E.; Browne, M.A.O. Obesity and mental disorders in the adult general population. *J. Psychosom. Res.* **2008**, *64*, 97–105.
8. Coetzee, V.; Chen, J.; Perrett, D.I.; Stephen, I.D. Deciphering faces: Quantifiable visual cues to weight. *Perception* **2010**, *39*, 51–61.
9. Pham, D.D.; Do, J.H.; Ku, B.; Lee, H.J.; Kim, H.; Kim, J.Y. Body mass index and facial cues in Sasang typology for young and elderly persons. *Evid. Based Complement. Altern. Med.* **2011**, *2011*, 749209. [CrossRef]
10. Wen, L.; Guo, G. A computational approach to body mass index prediction from face images. *Image Vis. Comput.* **2013**, *31*, 392–400.
11. Bipembi, H.; Panford, J.K.; Appiah, O. Calculation of body mass index using image processing techniques. *Int. J. Artif. Intell. Mechatron.* **2015**, *4*, 1–7.
12. Amador, J.D.; Cabrera, J.E.; Cervantes, J.; Jalili, L.D.; Castilla, J.S.R. Automatic calculation of body mass index using digital image processing. In Proceedings of the Workshop on Engineering Applications (WEA), Medellín, Colombia, 17–19 October 2018; pp. 309–319.
13. Madariaga, N.E.Q.; Noel, B.L. Application of artificial neural network and background subtraction for determining body mass index (BMI) in android devices using bluetooth. *Int. J. Eng. Technol.* **2016**, *8*, 366–370. [CrossRef]
14. Pons-Moll, G.; Baak, A.; Helten, T.; Müller, M.; Seidel, H.P.; Rosenhahn, B. Multisensor-fusion for 3d full-body human motion capture. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 663–670.
15. Ganapathi, V.; Plagemann, C.; Koller, D.; Thrun, S. Real time motion capture using a single time-of-flight camera. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 755–762.
16. Du, Y.; Wong, Y.; Liu, Y.; Han, F.; Gui, Y.; Wang, Z.; Kankanhalli, M.; Geng, W. Marker-less 3d human motion capture with monocular image sequence and height-maps. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.
17. Lee, K.; Lee, I.; Lee, S. Propagating LSTM: 3d pose estimation based on joint interdependency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 119–135.
18. Kwon, B.; Kim, D.; Kim, J.; Lee, I.; Kim, J.; Oh, H.; Kim, H.; Lee, S. Implementation of human action recognition system using multiple Kinect sensors. In Proceedings of the 16th Pacific Rim Conference on Multimedia (PCM), Gwangju, Korea, 16–18 September 2015; pp. 334–343.
19. Kwon, B.; Kim, J.; Lee, S. An enhanced multi-view human action recognition system for virtual training simulator. In Proceedings of the Asia–Pacific Signal and Information Processing Association Annual Summit Conference (APSIPA ASC), Jeju, Korea, 13–16 December 2016; pp. 1–4.

20. Kwon, B.; Kim, J.; Lee, K.; Lee, Y. K.; Park, S.; Lee, S. Implementation of a virtual training simulator based on 360° multi-view human action recognition. *IEEE Access* **2017**, *5*, 12496–12511. [CrossRef]
21. Lee, I.; Kim, D.; Lee, S. 3D human behavior understanding using generalized TS-LSTM networks. *IEEE Trans. Multimed.* **2020**. [CrossRef]
22. Wen, G.; Wang, Z.; Xia, S.; Zhu, D. From motion capture data to character animation. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST), Limassol, Cyprus, 1–3 November 2006; pp. 165–168.
23. Zhang, X.; Biswas, D.S.; Fan, G. A software pipeline for 3D animation generation using mocap data and commercial shape models. In Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR), Xi'an, China, 5–7 July 2010; pp. 350–357.
24. Zhuang, Y.; Xiao, J.; Wu, Y.; Yang, T.; Wu, F. Automatic generation of human animation based on motion programming. *Comput. Animat. Virtual Worlds* **2005**, *16*, 305–318. [CrossRef]
25. Kwon, B.; Huh, J.; Lee, K.; Lee, S. Optimal camera point selection toward the most preferable view of 3D human pose. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**. [CrossRef]
26. Gu, J.; Ding, X.; Wang, S.; Wu, Y. Action and gait recognition from recovered 3-D human joints. *IEEE Trans. Syst. Man Cybern. Part B* **2010**, *40*, 1021–1033.
27. Choi, S.; Kim, J.; Kim, W.; Kim, C. Skeleton-based gait recognition via robust frame-level matching. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2577–2592. [CrossRef]
28. Andersson, V.O.; Amaral, L.S.; Tonini, A.R.; Araujo, R.M. Gender and body mass index classification using a Microsoft Kinect sensor. In Proceedings of the 28th International Florida Artificial Intelligence Research Society (FLAIRS) Conference, Hollywood, FL, USA, 18–20 May 2015; pp. 103–106.
29. Kocabey, E.; Camurcu, M.; Ofli, F.; Aytar, Y.; Marin, J.; Torralba, A.; Weber, I. Face-to-BMI: Using computer vision to infer body mass index on social media. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), Montreal, Canada, 15–18 May 2017; pp. 572–575.
30. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC) 2015, Swansea, UK, 7–10 September 2015; pp. 1–12.
31. Mingqiang, Y.; Kidiyo, K.; Joseph, R. A survey of shape feature extraction techniques. *Pattern Recognit.* **2008**, *15*, 43–90.
32. Nahavandi, D.; Abobakr, A.; Haggag, H.; Hossny, M.; Nahavandi, S.; Filippidis, D. A skeleton-free Kinect system for body mass index assessment using deep neural networks. In Proceedings of the IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; pp. 1–6.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Reading B.D.; Freeman, B. Simple formula for the surface area of the body and a simple model for anthropometry. *Clin. Anat.* **2005**, *18*, 126–130. [CrossRef]
35. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
36. Elrahman, S.M.A.; Abraham, A. A review of class imbalance problem. *J. Netw. Innov. Comput.* **2013**, *1*, 332–340.
37. Ali, A.; Shamsuddin, S.M.; Ralescu, A.L. Classification with class imbalance problem: A review. *Int. J. Adv. Soft Comput. Appl.* **2015**, *7*, 176–204.
38. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
39. Van Hulse, J.; Khoshgoftaar, T.M.; Napolitano, A. An empirical comparison of repetitive undersampling techniques. In Proceedings of the IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, USA, 10–12 August 2009; pp. 29–34.
40. Ganganwar, V. An overview of classification algorithms for imbalanced datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
41. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
42. Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A. Real-time human pose recognition in parts from single depth images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011; pp. 1297–1304.

43. Araujo, R.; Graña, G.; Andersson, V. Towards skeleton biometric identification using the Microsoft Kinect sensor. In Proceedings of the 28th Symposium on Applied Computing (SAC), Coimbra, Portugal, 18–22 March 2013; pp. 21–26.
44. Andersson, V.; Dutra, R.; Araujo, R. Anthropometric and human gait identification using skeleton data from Kinect sensor. In Proceedings of the 29th Symposium on Applied Computing (SAC), Gyeongju, Korea, 24–28 March 2014; pp. 60–61.
45. Andersson, V.; Araujo, R. Full body person identification using the Kinect sensor. In Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Limassol, Cyprus, 10–12 November 2014; pp. 627–633.
46. Andersson, V.; Araujo, R. Person identification using anthropometric and gait data from Kinect sensor. In Proceedings of the 29th Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 425–431.
47. Yang, K.; Dou, Y.; Lv, S.; Zhang, F.; Lv, Q. Relative distance features for gait recognition with Kinect. *J. Vis. Commun. Image Represent.* **2016**, *39*, 209–217. [CrossRef]
48. Huitzil, I.; Dranca, L.; Bernad, J.; Bobillo, F. Gait recognition using fuzzy ontologies and Kinect sensor data. *Int. J. Approx. Reason.* **2019**, *113*, 354–371. [CrossRef]
49. Sun, J.; Wang, Y.; Li, J.; Wan, W.; Cheng, D.; Zhang, H. View-invariant gait recognition based on Kinect skeleton feature. *Multimed. Tools Appl.* **2018**, *77*, 24909–24935. [CrossRef]
50. Kwon, B.; Lee, S. Human skeleton data augmentation for person identification over deep neural network. *Appl. Sci.* **2020**, *10*, 4849. [CrossRef]
51. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
52. Hsu, C.W.; Chang, C.C.; Lin, C.J. *A Practical Guide to Support Vector Classification*; National Taiwan University: Taipei, Taiwan, 2003.
53. Wang, Q.; Kurillo, G.; Ofli, F.; Bajcsy, R. Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect. In Proceedings of the IEEE International Conference on Healthcare Informatics, Dallas, TX, USA, 21–23 October 2015; pp. 380–389.
54. Albert, J. A.; Owolabi, V.; Gebel, A.; Brahms, C. M.; Granacher, U.; Arnrich, B. Evaluation of the pose tracking performance of the Azure Kinect and Kinect v2 for gait analysis in comparison with a gold standard: A pilot study. *Sensors* **2020**, *20*, 5104. [CrossRef] [PubMed]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.