Gait Abnormality Detection Using Multimodal Sensors and Machine Learning

Fritz Kevin Flores
De La Salle University
Manila, Philippines
fritz.flores@dlsu.edu.ph

Adrian Charles Mendoza
De La Salle University
Manila, Philippines
adrian_mendoza@dlsu.edu.ph

Antonio Enrico Roque
De La Salle University
Manila, Philippines
antonio_roque@dlsu.edu.ph

Anna Francesca Santos
De La Salle University
Manila, Philippines
anna_francesca_santos@dlsu.edu.ph

Vivian Mae Uy
De La Salle University
Manila, Philippines
vivian_uy@dlsu.edu.ph

***Abstract*—Gait can be defined as how a person walks. When the person is not able to walk properly due to different factors it can be deduced that their gait is abnormal. The proponents want to simplify and improve the process of detecting whether a person has gait abnormality or not through the use of technologies. In this study, a Kinect sensor and a smart flooring sensor were used together to collect the data needed for the study. Overall, the researchers had 30 samples containing 20 participants with an additional of 10 acted data. Features such as Stride length, Symmetry, and Cadence were collected. Then, the data were preprocessed by using methods such as feature scaling, feature extraction, and feature selection before being fed into K-Nearest Neighbors (KNN) and K-means Clustering machine learning algorithms, as well as Convolutional Neural Network (CNN) deep learning algorithms, to create different classifiers for gait. Finally, evaluation metrics such as accuracy, DBI, and loss were used to select the best classifiers that would be used in the decision-level fusion.**

***Index Terms*—gait abnormality detection, machine learning, decision-level fusion, multi-sensor, Kinect, FSR**

I. Introduction

The ability to walk is crucial for human mobility and useful for predicting the quality of life [1]. Every person walks differently and the manner of how a person walks is called gait [2]."Normal gait is both stable and flexible, allowing for changes in speed and maneuvering in different terrains while maintaining energetic efficiency" as defined by [3]. If the person cannot walk properly due to illness, genetic factors, injuries or abnormalities in the legs or feet, then the gait of that person is considered abnormal [4]. Since walking is the most basic mode of transportation for humans according to [5], the inability to walk can affect the person drastically. Moreover, if gait disorders are detected and assessed in time for treatment, it can help prevent future complications that can arise from the disorder as mentioned by [6]. In order to determine the walking patterns of a person, gait analysis is being utilized. Currently, in a clinical setting, gait analysis is done through qualitative means such as human observations, visual assessment, and self-reporting of patients [7]. Even though gait abnormality can be observed, some subtle changes can be overlooked and is prone to human error. The main objective of this research is to apply machine learning to different types of sensor data, obtained from a self-made low-cost smart setup, which will be used to detect gait abnormality. Together with the main objective, the study aims to build a low-cost smart flooring to be able to detect pressure data, create a program that will collect data from the smart flooring and vision camera sensors, implement preprocessing techniques to the data collected, and create a binary machine learning classifier of normal and abnormal gait.

II. Review of Related Literature

Multiple studies related to gait analysis have been conducted over the years. Three sensors that have been repeatedly used in those studies are (1) vision camera, (2) flooring and (3) wearable sensors.

[8] presented a computer vision approach for gait analysis to detect early frailty and senility syndromes with the use of Android smartphone cameras together with OpenCV to record and process a gait sequence in obtaining spatiotemporal features. These features were then sent to the cloud for classification between normal and abnormal gait. K-nearest neighbor (KNN) algorithm with Dynamic Time Warping (DTW) was implemented and stride and leg-angle time series were used to classify the gait. Their results proved that sagittal view produces more accurate results than frontal view given that sagittal results range from 82%-93% while frontal results range from 72%-89%. The research of [9] is another study that utilized vision camera sensors, more specifically a Kinect camera, to conduct gait analysis. Their objective is to execute a geometric model-based algorithm in order to process the infrared and depth image sequences and skeleton joint points that they obtained from the Kinect. Through the processed data, movement patterns will be produced regardless of whether the background is complex or not. When compared to the VICON, which is a commercial product that captures motion, the accuracy rate of this research is higher.

As for the flooring sensors, [10] used the Tekscan® Flexiforce pressure sensors to obtain force distribution and pressure points but it was expensive costing over 500,000 Pesos for a setup of 512 sensors. Aside from the pressure sensors, their setup also included a Kinect V1 which served as a tool for cross validation and an accelerometer which collected acceleration data. Overall, the setup is both intrusive and expensive. The data collected is fed to a polynomial regression model to foretell and spot abnormal behaviour. Same pressure sensors as the previous one were also applied in the work of [11]. However, they placed it in a smart shoe. The smart shoe determines the type of gait a person has and sends its diagnosis to a hand-held device via Bluetooth. The data is processed and converted using complementary and Kalman filtering techniques. The output from these filters were used as a basis to detect the type of gait. The accuracy score of Kalman filtering technique was higher than the score of complementary filtering technique.

Another example of a wearable sensor is in the study of [12], wherein in this case, an intrusive wearable camera is attached on the leg of the person with the camera pointing downward to capture the walking motion, including the position and pose of the person. Extended Kalman Filter was used in predicting the walking state of the subject which was categorized into 5 states such as walking slow, walking normally, running, turning right, or running left. The results of the study indicated a match in the initial visual classification on the walking speed and even when the feature points were not detected and the walking motion was different from the stored motion, the estimation was stable.

The research of [13] made use of 1 force sensitive resistor and 2 accelerometers. Using the said sensors, their goal was to detect real-time gait events of slow, normal, and altered walking. FSR algorithms (FSR Force, FSR derivative) and accelerometer algorithms (AccA) were utilized in measuring the accuracy of detection. The results show that the FSR system showed significantly lower errors than the accelerometer system, while both systems had increased accuracy compared to previously reported real-time ambulatory systems.

Using a wearable inertial measurement unit (IMUs) to detect gait abnormality in subjects with neurological disorders, the study of [14] had different patient groups (control groups, stroke patients, convalescence patients) in order to separate those who have gait abnormalities. Gait features including spatio-temporal parameters were selected and a continuous assessment was done to check for any indications of improvement or deterioration of the lower limb. Their research made use of different algorithms such as ellipsoid fitting which was used to deal with local magnetic disturbance, Hidden Markov model (HMM) and Kalman Filter were employed to illustrate the gait model and eliminate false gait phase partition while Zero Velocity Updates algorithm is adopted as pseudo observable to eliminate integral errors.

However, with all the different studies and sensors mentioned, most of the studies are expensive and focused mainly on detecting gait using intrusive sensors or only vision sensors. As observed from the different researches, it can be deduced that there is a research opportunity on technology which incorporates a decision-level fusion of model results using data from different non-intrusive sensor-types while being low-cost.

III. Methodology

A. Overview

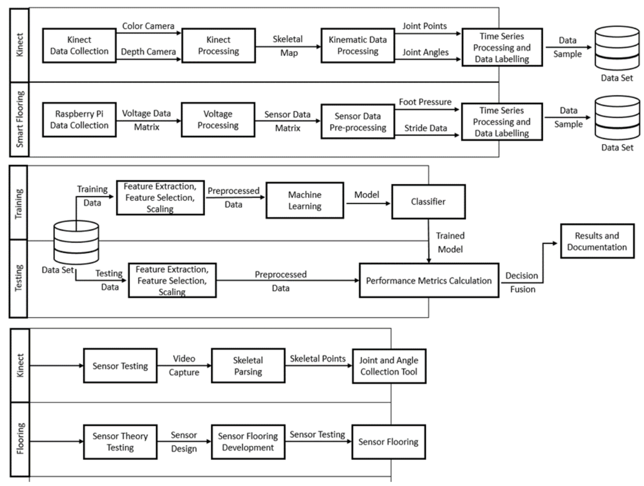

Fig. 1. Flowchart of each Methodology Framework

The steps that were done in achieving the objectives of the study are divided into three major phases namely (1) tool development, (2) data collection, and (3) machine learning and testing. The tool development comprises the creation of the smart flooring, the Kinect setup, and the design of the programs that were used to collect the data. The data collection step includes the process of collecting data from the participants using the Kinect sensor and the smart flooring, cleaning, and labeling the data. Lastly, the machine learning and testing step explained the different preprocessing techniques applied to the data collected before it was fed into a model, and the performance metrics used to evaluate the model.

B. Creation of Smart Flooring

A low-cost flooring was developed by following the idea of how pressure sensitive flooring works while using inexpensive conductive materials. Materials such as Copper Tape, Velostat, Tarpaulins, and wires were used in developing the smart flooring. Raspberry Pi and microcontrollers were then used to read the data.

The self-made smart flooring consists of 3 layers as seen in Figure 2. At the very bottom is a 160x60cm tarpaulin with 15 vertical strips of copper tape as seen in figure 4. Next is 10 pieces of Velostat, each with a measurement of 20x20cm, placed in a 5 by 2 tiles manner. After the Velostat, another layer of tarpaulin is placed with 40 horizontal strips of copper tape instead as seen in figure 3. This is to achieve the 600 sensors of the smart flooring.

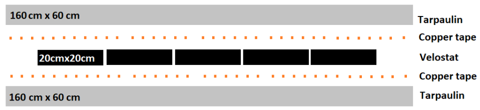

Fig. 2. Layers of the Smart Flooring

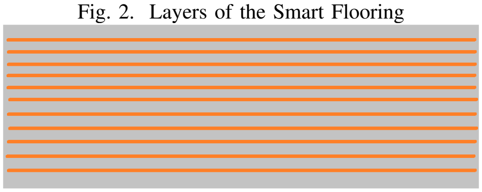

Fig. 3. Top view of First Layer of Tarpaulin with Copper Tape

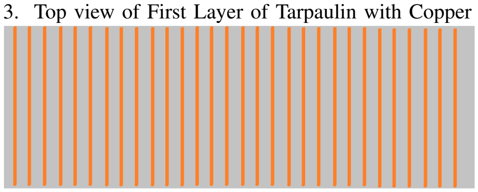

Fig. 4. Top view of Second Layer of Tarpaulin with Copper Tape

C. Kinect Setup

The Kinect was placed at the side of the smart flooring as seen in Figure 5, with 103 inches away from the smart flooring. The Kinect version 2 has a horizontal field of view of 70 degrees and a vertical field of view of 57 degrees [15]. By putting the Kinect in this area, important data such as angles formed by the lower body joints, and spine movement were captured while the subject was walking. The researchers also tested other locations and it turns out that putting the Kinect at the side was the best placement to gather the data the research needs.

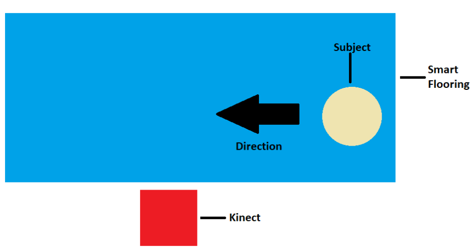

Fig. 5. Top View of the Environment Setup

IV. Data Collection

A. Procedure

The data collected were from two kinds of participants: (1) with gait abnormality, and (2) without gait abnormality. All participants must be adults of legal age. Before the data collection, the participants have undergone an assessment to check if they are in good physical condition else the collection would be canceled or postponed to another date. They were also asked to sign an informed consent form before proceeding to the data collection. Upon agreeing, the participants would then walk on the environment setup 10 times from right to left. 10 rounds were done in order to collect the data more precisely, as well as, to get a general average of the actual gait of the person. The researchers made the environment setup as natural as possible to prevent the participants from being conscious during data collection. After gathering enough data from different participants, a physical therapy student was asked to diagnose and interpret the data, a video where the participants can be seen walking in their usual way, in their own manner to label which data would result in a normal and abnormal gait. It resulted with 15 normal data and 5 abnormal data. The researchers acted out an additional 10 abnormal gait data, which were labeled as abnormal by the physical therapy student, in order to balance the dataset having 15 normal and 15 abnormal data. In total, the researchers had 30 samples containing 20 participants with an additional of 10 acted data.

B. Raspberry Pi Data Collection

When collecting data from the smart flooring, features such as pressure points, position, and time were considered. The pressure points, initially ranging from values 0 to 1023, were first transformed to values ranging from 0 to 100 with the use of the formula below:

$$initialvalue * 100/1023 \tag{1}$$

These values determine which areas of the smart flooring were stepped upon by the participant and the pressure applied in those areas. A value of 0 indicates no pressure while a value of 100 indicates the strongest pressure. The location of each footstep on the smart flooring was also taken into account as it helped determine the gait of a person. The timestamp of each step was recorded in the format of mm:ss.f to be used to get other important features like step and swing time. Figure 6 is a sample outline of the foot pressed on the flooring. These data collected help describe the gait of a person. When people walk, the pressure and location of each foot step as well as the time it takes for them to complete a gait cycle vary from person to person. Some walk faster than the others which makes the time shorter, other people do not walk on the same path consistently which fluctuates the location and, there are also people who do not balance well when walking which may be caused by the uneven distribution of pressure of their footstep. These data play a huge role in gait analysis because combining all of these features make the gait of a person unique. It also helps determine if a person is walking abnormally.

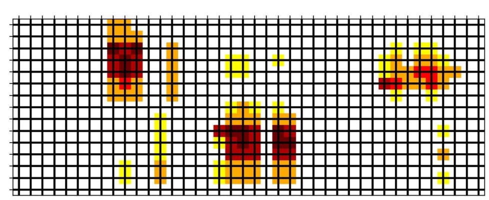

Fig. 6. Sample Outline of the Footsteps Pressed on the Flooring

The smart flooring was connected to an analog to digital converter to convert the analog signals read from the sensors to numerical values. A python program was run in the Raspberry

Pi to save the transmitted data in a CSV file while the collection is happening. The program iterates through all the columns and rows of the entire smart flooring, sending electricity to the row in question, while reading the value at the columns. It was able to save around 22 to 23 samples per second. When saving to the csv file the 40 by 15 grid were represented as rows 0-39 and columns 0-14 which created a 2D matrix. Each tile in the 2D matrix was then named based on the row number and the column number, for example, row0_column0 up to row39_column14.

C. Kinect Data Collection

In order to capture the data coming from the Kinect sensor, the researchers created a program using the Python programming language and libraries. Three kinds of data were collected: (1) coordinates of each joint, (2) the length between these joints, and (3) the angles that these joints form. When a person walks, each of his body joints move continuously. Then the joints next to each other form an angle and as the joints move, these angles consequently change as well. For example, the arms of a person sway as he walks, which makes the wrist, elbow, and shoulder joints form angles that constantly change from 90 degrees (an L shape) to 180 degrees (a straight line). The only consistent feature is the length between each joint and these lengths impact the walk of a person too. For instance, a person with relatively shorter legs takes more steps than a person with longer legs when walking the same distance [16]. Similar to features collected by Raspberry Pi, these features are important in the process of gait analysis because it differentiates a person from the others and it helps detect gait abnormality.

In obtaining these data, the PyKinect library was used. This library provides a straightforward way to get the coordinates of the joints since it automatically stores the data in an array called jointPoints. As for the length between the joints, it was computed using the distance formula as seen in Figure 7. Lastly, the code in Figure 8 shows how the angles were calculated by using functions math.atan2 and math.degrees where the first line will obtain the numeric value between negative pi and pi that represents the angle of a point and positive x-axis. The second line, on the other hand, converts the angles in radians to degrees. When the angle is negative, 360 was added to the angle to keep the results between 0 and 360 degrees only.

```
def compute_length (self, x1, y1, x2, y2):
    return math.sqrt((x1 - x2)**2 + (y1 - y2)**2)
```

Fig. 7. Sample Code of Getting Length between the Joints

```
def getAngle(self, x1, y1, x2, y2, x3, y3):
    ang = math.degrees(math.atan2(y3-y2, x3-x2) - math.atan2(y1-y2, x1-x2))
    return ang + 360 if ang < 0 else ang
```

Fig. 8. Sample Code of Getting Angles formed by Joints

When saving to the csv file, for the x and y coordinates, it is simply named as the body part then x or y, for example, "head_x". As for the length, it is named using the parts of the body part as points of the line segment for instance, "len_spineshoulder_spinemid" would indicate the length from the point in the spine shoulder until the middle of the spine. As for the angles, it is named according to the points of an angle for example "ang_shoulderright_elbowright_wristright", refers to the points of the angle with elbowright as its vertex.

D. Cleaning of Data

For the smart flooring data collection, the data tiles within the duration of one walking process were then combined together in one row in such a way that one data entry would consist of all the points from its initial contact until they complete the whole walk. Since each participant's walking time varies, in order to normalize the data collected from each person, each sample was split into 10% intervals, to obtain a total of 10 intervals representing one sample by getting the min, max, and ave from each interval. Aside from the force in each row and column coordinate, the following were also extracted from the raw data: (1) center of force, (2) step time, (3) swing time, (4) stride length, (5) weight distribution, (6) gait symmetry, and (7) cadence. The sample screenshot of this csv file is seen in Figure 9. The proponents also created another version of the dataset by removing data that contains 0 stride length or negative values for gait symmetry and cadence since it does not make sense for a person to have 0 or negative values in those features.

	Label	ID	timestamp	StrideLength	StepTime	SwingTime	Symmetry	Cadence	Right_CoF_row	Right_CoF_col	...	10_Row39_Column11_ave
0	1	1	3.1	38	0.7	0.8	12.048193	74	1	1	...	0
1	1	2	3.5	40	0.7	1.0	2.409639	60	1	1	...	0
2	1	3	3.6	38	0.1	1.0	10.000000	84	2	5	...	0
3	1	4	3.3	32	0.6	0.7	9.756098	84	0	1	...	0
4	1	5	3.8	36	0.6	1.0	19.753086	84	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
265	0	266	3.0	0	1.0	-1.0	-1.000000	-1	1	1	...	0
266	0	267	4.3	42	1.0	1.1	8.187135	66	0	4	...	0
267	0	268	3.4	42	0.9	1.1	17.283951	48	0	4	...	0
268	0	269	3.2	40	0.9	1.2	3.636364	74	1	0	...	0
269	0	270	3.3	42	0.5	0.9	3.550296	50	0	5	...	0

Fig. 9. Screenshot of the Smart Flooring Data CSV File

As for the Kinect data collection, the data collected is 15 frames per second thus there are many instances per ID as well. The process of combining applied to the Kinect data is also the same as the smart flooring data as multiple instances of the same ID were appended to produce a single entry per walk. A sample database of the merged data with respect to its time interval is shown in Figure 10.

	Label	ID	time_stamp	head_x_min	head_x_max	head_x_ave	head_y_min	head_y_max	head_y_ave	neck_x_min
0	1	1	3.9	651	690	670.000000	263	270	266.200000	625
1	1	2	3.9	636	707	667.600000	261	279	266.600000	630
2	1	3	3.5	674	761	712.200000	268	287	275.600000	661
3	1	4	3.8	649	715	681.400000	265	273	269.400000	640
4	1	5	4.1	594	640	615.500000	256	258	256.833333	597
...	...	...	...	...	...	...	...	...	...	...
265	0	266	4.2	702	716	707.166667	430	432	431.500000	676
266	0	267	4.2	671	694	680.333333	432	435	433.166667	646
267	0	268	4.1	660	700	678.000000	423	427	425.166667	636
268	0	269	3.8	650	703	674.200000	439	444	442.400000	685
269	0	270	4.0	703	773	736.333333	428	434	432.000000	674

Fig. 10. Screenshot of the Kinect Data CSV File

V. Machine Learning and Testing

Once the data was collected and labeled accordingly by the physical therapy student, machine learning was implemented to help determine if the person has a gait abnormality or not. The training set was fed into numerous machine learning algorithms for various possible models. For supervised algorithms, the correct labels were included in the training set, while for the unsupervised, the labels were not given. The algorithms that performed the best among all the others based on evaluation metrics such as accuracy would be used for decision level fusion

A. Preprocessing of Data

Before feeding the data to the machine learning algorithm, the proponents applied various machine learning techniques to better improve the data. First, feature scaling was performed to normalize the data in a particular range, so that features with higher values will not be assumed to have superiority over other features. This matters especially when dealing with classifiers that compute distance. In this research, three scaling methods were used, namely Standard Scaler, MinMax Scaler, and Normalization from the sklearn library, to test out which produces the best results. After scaling was done to the data, feature extraction was applied to it. Feature extraction reduces the dimension of the features to a more manageable number of features for processing by creating new data, selected and/or combined from the raw data, while still representing the original data accurately. This is helpful when dealing with a dataset that has redundant data. The following are the feature extraction techniques used: Principal Components Analysis, Independent Component Analysis, and Locally Linear Embedding from sklearn. The proponents have also considered using feature selection methods such as Feature Importance. Feature importance calculates the importance of a feature and gives out scores. This is used by the researchers to select the best top n features that would give the best result. The range of n varies depending on the number of the features.

Different kinds of dataset were fed into the models for testing:

- The original dataset without applying scaling, feature extraction, and feature selection.
- The different datasets produced after applying different combinations of feature selection and feature extraction methods to each scaling method discussed.

For smart flooring data, additional datasets which include only the gait features extracted from the raw data was created. This is done to both of the datasets, all and cleaned versions. The preprocessing methods discussed were also applied to them.

The datasets mentioned were fed to K-Nearest Neighbors, K-means Clustering, and Convolutional Neural Network algorithms.

B. K-Nearest Neighbors

K-Nearest Neighbors is a kind of supervised machine learning algorithm, meaning labels are given to the model. This algorithm considers k numbers of neighbors or data points closest to the data to predict its class; the majority class wins. In this research, the dataset was split into training (80%) and testing (20%) sets. The random_state was set to 42 to ensure that the same data will be used again for testing purposes. In choosing the optimal number of k, values from 1 to 10 were considered. The number that yields the best accuracy will be the final k to be used in the model. K-fold validation was also implemented to cross validate the results of the machine learning model in order to achieve an unbiased model. The range of the 2 to 10 was used in choosing the best number of folds for K-fold. The results of the algorithm were measured through their accuracy values wherein the higher the value, the better.

C. K-means Clustering

K-means Clustering is an unsupervised machine learning algorithm in which it groups the given data into clusters that have the same characteristics. These data are grouped together in clusters because of certain similarities with its features. For K-means clustering, the components parameter for all of the feature extraction methods were set to k = 2. Then, the proponents made use of the Elbow Method to determine the ideal number of clusters to be formed from the data. In evaluating the performance of the algorithm in terms of clustering, Davies Bouldin Index was performed. This would determine how well the clustering has been done. The lower the value of Davies Bouldin Index, the better the result is.

D. Convolutional Neural Networks

CNN is a type of deep learning that commonly deals with problems that involve visual representation inputs. In this study, it was implemented using a transfer learning technique where a pre-trained model, Inception v3 from Tensorflow, was used in training the new dataset. This is to reduce the time of training since the pre-trained model was already trained with large datasets using high power GPUs. Since this approach requires data in image form, the dataset was composed of images generated from the preprocessed datasets, both Kinect and smart flooring. They were then split into a train set (80%) for training the model, a validation set (10%) for tuning, and a test set (10%) for evaluating the model. The proponents had freezed all the layers except the last five similar to what other sources have done in order to allow modifications. The image was preprocessed using ImageDataGenerator from Keras. ReLu and sigmoid were used for the activation functions, then Binary Cross Entropy for the loss metric since this study is dealing with binary classification problem. Additionally, a dropout rate of 0.2 was used to minimize overfitting, and a callback function was created to stop the training once the model has reached 99.9% of accuracy. For the optimizer, RMSprop with a learning rate of 0.0001 was considered. After creating the model, it was trained, validated, and tested. The training set has two types as well, one with data augmentation and the other without. The number of epochs is set to 50 considering the amount of training time it takes. The proponents also tried another approach where not all the layers from the pre-trained model were used and the rest is just similar to the first approach with all the layers. The model is then evaluated using accuracy and loss.

VI. Decision-level Fusion

Once the results of the machine learning algorithms were available, the highest accuracy of each algorithm was compared. The researchers got the highest accuracy produced from both the Kinect and the smart flooring machine learning results, then the model that produced the highest accuracy will be selected for the output and weights utilized in the decision-level fusion. The highest accuracy would serve as the weight multiplier for the decision level fusion using the formulas below.

$$KinectWeight = \frac{KinectAccuracy}{KinectAccuracy + MatAccuracy} \tag{2}$$

$$MatWeight = \frac{MatAccuracy}{KinectAccuracy + MatAccuracy} \tag{3}$$

With the use of the weights and the kinect and smart flooring results obtained from the chosen models, the decision-level fusion was implemented using the formula below:

$$\begin{aligned} Result = (KinectWeight \times KR) + \\ (SmartFlooringWeight \times SR) \end{aligned} \tag{4}$$

Wherein the value of Kinect Result (KR) and Smart Flooring Result (SR) corresponds to a prediction of 1 or 0. The resulting value would correspond to the confidence of the predictions for both Kinect and smart flooring. A threshold value of 50% would be used wherein a value greater than 50 corresponds to an abnormal gait while a value less than 50 corresponds to a normal gait.

VII. Results and Analysis

A. K-Nearest Neighbors

Accuracy	Not Preprocessed	Standard Scaler	MinMax Scaler	Normalized
w/o feature extraction	92.59	90	90	89
PCA	92.59	90	90	90
ICA	81	88	90	84
LLE	80	72.96	76	80

Legend:
With Feature Importance, W/o Feature Importance, Both
Lowest Accuracy

Fig. 11. KNN Kinect Data

As shown in Figure 11, the values that are highlighted in red means that it was run on a feature selection test where the algorithm calculated the feature importance ratios and then ran it based on those results. On the contrary, values that are highlighted in green mean that they were not run on feature importance results and lastly the values that are not highlighted means that results for both before and after feature selection were the same. Upon viewing figure 1, it could be assumed running feature importance selection would not give us better results considering that only 3 out of 16 values are better when run through feature importance selection. On the other hand, only 2 out of 16 results are better when run without feature importance selection. The rest of the values which account for 11 out of 16 of the results are both equal whether or not feature importance selection was run. It can also be seen that the highest results come from the original data where no feature scaling was done in addition to no feature extraction as well as the feature extraction of PCA both at 92.59%. On the contrary, the lowest result came from the standard scaler feature scaling together with the LLE feature extraction process which resulted at 72.96%. It can be noticed that ICA and LLE feature extraction processes produce lower results as compared to PCA and not using any feature extraction process possibly because PCA tries to find an orthogonal linear transformation while ICA tries to find a linear transformation.

Accuracy	Not Preprocessed	Standard Scaler	MinMax Scaler	Normalized
w/o feature extraction	84	81	87	84
PCA	86	78	87	85
ICA	85	85	84	86
LLE	**80**	**77.09**	87.94	85

Legend:
With Feature Importance, W/o Feature Importance, Both
Original, Cleaned, *Same*
Lowest Accuracy

Fig. 12. KNN Smart Flooring All Data

Accuracy	Not Preprocessed	Standard Scaler	MinMax Scaler	Normalized
w/o feature extraction	81.02	**75**	**76**	**78**
PCA	**78**	**75**	**76**	**78**
ICA	**76**	73	**74**	**73**
LLE	72	60.69	68	71

Legend:
With Feature Importance, W/o Feature Importance, Both
Original, Cleaned, *Same*
Lowest Accuracy

Fig. 13. KNN Smart Flooring Gait Features Data

As seen in Figures 12 and 13, values highlighted in red means that the best result was gathered after running feature importance selection while the data highlighted in white meant that running the data with or without feature importance selection would produce the same results. On the other hand, values highlighted in green means that it was retrieved without feature importance selection. In addition, values on bold meant that they were retrieved from the original data results while the values that are italicized meant that they were retrieved from the cleaned data results.

Figure 12 shows that feature importance selection produces better results compared to not adding it through the process where it gave us 6 out of 16 of the results. On the other hand, 2 out of 16 of the results are achieved without running feature importance selection, the rest of the data are then achieved whether or not feature importance selection was run which accounts for half or 8 out of 16 of the data. In addition, Figure 12 gives us a lowest accuracy of 77.09% when running standard scaler feature scaling while on the LLE feature extraction on the original dataset. On the contrary, the highest result of accuracy 87.94% was achieved on the cleaned dataset while running on the minmax feature scaling and the LLE feature extraction. It is also noticeable that most of the highest accuracies came from using the clean dataset, which means that removing outliers and values with -1 greatly improved the algorithm to perform better. These -1 values could be serving as noise rather than substantial data for KNN.

On a contrast to Figure 12, Figure 13 shows that feature importance selection does not give better results where only 2 out of 16 results are better as compared to getting 5 out of 16 of the results without running feature importance selection and the remaining 9 out of 16 of the results produced the same accuracies. In addition to the percentages, the highest accuracy was retrieved when running the algorithm on the cleaned dataset while not using any feature extraction and feature scaling which produced a result of 81.02%. On the flip side, the lowest accuracy was retrieved when running the cleaned dataset while running on the standard scaler feature scaling algorithm and the LLE feature extraction process which got a 60.69% accuracy. In this aspect, using a data set of mainly gait features, the original data before cleaning performed better. This could be interpreted that the -1 values are an important signifier for KNN to determine. It could be a basis to determine the gait abnormality of a person because upon observation of the data, most abnormal data have -1 values in its features.

B. K-Means Clustering

DBI	Not preprocessed	Standardized	Normalized
PCA	0.8025	0.9615	**0.7077**
ICA	0.7598	0.8491	**0.6387**
LLE	**0.2233**	0.3699	0.6224
Legend: Lowest DBI of each preprocessing method **Lowest DBI of each feature extraction**			

Fig. 14. DBI of Kinect Data

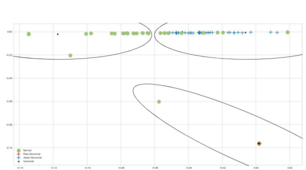

Fig. 15. Graph of Kinect Data labels with clusters using LLE without any preprocessing

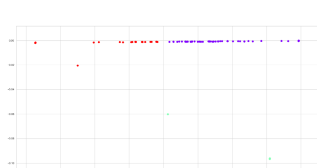

Fig. 16. Clusters graph of Kinect Data using LLE without any preprocessing

The best combination of feature extraction and preprocessing method to apply in K-means using the Kinect data is LLE without any preprocessing. It yielded the lowest DBI with a value of 0.2233 as seen in Figure 14. After applying these two methods to the Kinect data, K-Means was able to cluster the data into 3 which was a value selected by the elbow method. This means that it was able to identify 3 different walking patterns amongst the 30 gait instances. The initial goal of K-Means was to originally cluster the data into two: normal and abnormal. However, looking at Figure 15, it is noticeable that there are abnormal and normal points that are overlapping which means that there may be instances where the walking patterns of the normal and abnormal were similar which caused them to be clustered together. Upon checking the ID of the overlapping abnormal and normal data points, it was discovered that these instances were taken from the same person except that in some of the instances, the person was just acting out a gait abnormality. Despite the overlapping points, it can be observed in Figure 16 that the purple cluster covers some of the abnormal instances and the red cluster covers most of the normal instances. Also, even if the elbow method resulted in 3, the blue cluster only consists of outliers which are the real abnormal data according to Figure 15.

DBI	Not preprocessed	Standardized	Normalized
PCA	0.5614	0.1983	0.4195
ICA	0.6092	0.1995	0.5398
LLE	0.3199	**0.1682**	0.1991
Legend: All data, All (Gait Features Only) data **Lowest DBI**			

Fig. 17. Summary of the Best DBI of ALL Smart Flooring Data

The All data and All (Gait Features Only) data were summarized into one table as seen in Figure 17. This table was obtained by comparing the DBI values output by both the All data and the All (Gait Features Only) data then the DBI with lower values were placed in this table. Based on this summarized table, it can be concluded that the Gait Features Only data returned better values than the All data. This means that the gait features are good enough and that the coordinates and the raw pressure points of the foot steps are not significant features for K-Means.

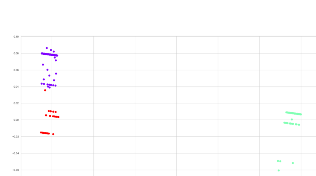

Fig. 18. Clusters graph of All (Gait Features Only) data using LLE with standardization

Fig. 19. Graph of All (Gait Features Only) data labels with clusters using LLE with standardization

The best combination of feature extraction and preprocessing method for both All and All (Gait Features Only) data is

LLE with standardization. It yielded a value of 0.1682 which is the lowest in Figure 17. When applied to the All (Gait Features Only) data, K-Means was able to cluster the data into 3 with the help of elbow method which is similar to the Kinect data results. Setting Figures 18 and 19 side by side, it is obvious that K-Means was not able to group the data that well mostly because of the overlapping data once again. Although the red and violet clusters were somehow able to differentiate the normal and abnormal data points, the overlapping points were still misgrouped. The overlapping points were checked via ID and it was discovered that most of the abnormal data points that overlapped with the normal data points were the real ones as seen in Figure 19 which is inconsistent with the results of Kinect data in Figure 15 because there, the abnormal points that overlapped with the normal points were the ones acted out.

DBI	Not preprocessed	Standardized	Normalized
PCA	0.6045	0.5801	0.6403
ICA	**0.5752**	0.6696	0.6469
LLE	0.6675	0.5766	0.6367
Legend: Clean data, Clean(Gait Features Only) data **Lowest DBI**			

Fig. 20. Summary of the Best DBI of CLEAN Smart Flooring Data

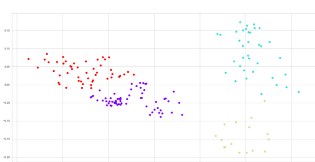

Fig. 21. Clusters graph of Clean data using ICA without any preprocessing

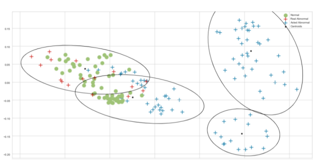

Fig. 22. Graph of Clean data label with clusters using ICA without any preprocessing

Similar to the All data, both the Clean and Clean (Gait Features Only) data were summarized into one table as seen in Figure 20. Based on the summarized table, it can be deduced that the Clean data has better DBI values than the Clean data. This may mean that Clean (Gait Features Only) data may not be enough for K-Means since it has the smallest size out of all four smart flooring datasets. Therefore the best combination feature extraction and preprocessing method to use for the Clean data based on the results after removing the outliers is ICA without any preprocessing. By applying this combination, the elbow method was able to output a K of 4 which means that K-Means grouped the Clean data into 4 as seen in Figure 21. In Figure 22, it can be observed that the overlapping data points are more severe. Also, similar to the results of All data, the overlapping abnormal points are the real ones. Almost all of the graphs that were obtained from the Clean and Clean (Gait Features Only) data looks like this. Therefore, it can be inferred that the data removed from the All and All (Gait Features Only) to create the Clean and Clean (Gait Features only) data makes the normal and abnormal less distinguishable from each other. The negative values probably help K-Means to determine that an instance is abnormal.

DBI	Not preprocessed	Standardized	Normalized
PCA	0.5614	0.1983	0.4195
ICA	0.5752	0.1995	0.5398
LLE	0.3199	**0.1682**	0.1991
Legend: From Summary Best of All From Summary Best of Clean			

Fig. 23. Overall Best DBI of ALL and CLEAN Smart Flooring Data

To summarize the results of the smart flooring data using K-Means, the best All data DBI in Figure 17 and the best Clean data DBI in Figure 20 were compared and summarized into one table as seen in Figure 23. It can be concluded that All (Gait Features Only) data is the overall best data to use as it yielded most of the lowest values. This means that the values removed from Clean data do have impact on the results but the raw pressure points and coordinates of each footstep may be insignificant to K-Means. Lastly, in terms of clusters, All (Gait Features Only) data's elbow method results were still better as it ranged from 3 to 5 in contrast to the elbow method results of Clean data that ranges from 4 to 6. With the right combination of feature extraction and preprocessing methods, K-Means is able to group the All (Gait Features Only) data better as well.

C. Convolutional Neural Network

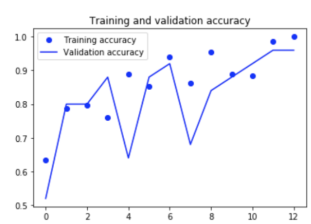

Fig. 24. Accuracy of the Model Trained Without Augmenting the Training Set

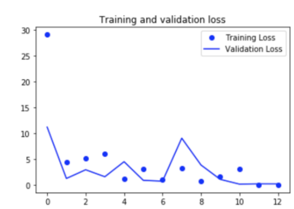

Fig. 25. Loss of the Model Trained Without Augmenting the Training Set

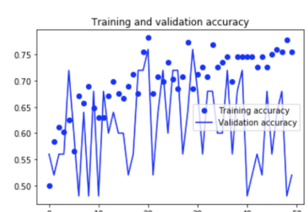

Fig. 26. Accuracy of the Model Trained With Augmented Training Set

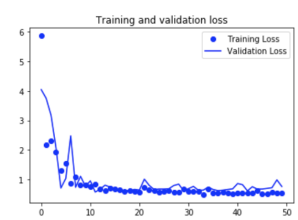

Fig. 27. Loss of the Model Trained With Augmented Training Set

The graphs shown are the results of using the Kinect dataset alongside with the model using all the layers in the pre-trained model. Figures 24 and 25 are the results of the model without applying data augmentation to the training set while Figures 26 and 27 were trained with augmented data. As observed in the graphs, the model without augmented training data has achieved higher validation accuracy and continued to improve, unlike the one with data augmentation. This could mean that data augmentation might have created data that have big differences to the original data, thus lower the accuracy. The model of Figures 24 and 25 was able to achieve 96% of validation accuracy and 76% for the model of Figures 26 and 27.

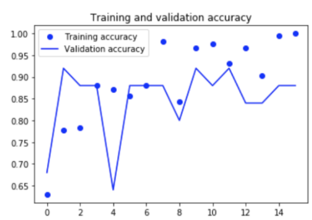

Fig. 28. Accuracy of the Model Trained Without Augmenting the Training Set

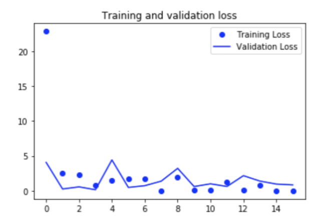

Fig. 29. Loss of the Model Trained Without Augmenting the Training Set

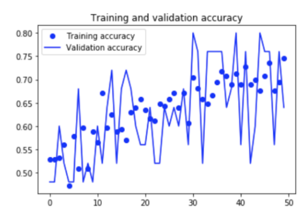

Fig. 30. Accuracy of the Model Trained With Augmented Training Set

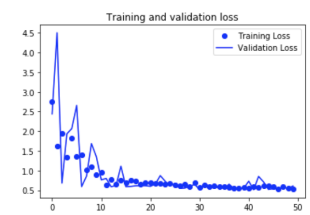

Fig. 31. Loss of the Model Trained With Augmented Training Set

On the other hand, here are the results of not using all the layers from the pre-trained model. It can be seen that the performance of the model increased compared to using all layers in the pre-trained model. This shows that not all the layers in the pre-trained model are necessary. Similarly, the model without data augmentation still produced higher validation accuracy. Highest validation accuracy attained was 92% for the model without data augmentation, and 80% for the model with augmented training set.

KINECT				
ALL Layers				
Train Dataset	Validation Dataset	Test Dataset	without data aug	with data aug
Original	Original	Original	96.43%	78.57%
NOT ALL Layers				
Original	Original	Original	100.00%	82.14%

Fig. 32. Accuracy of the Model Using Test Sets

After creating the model, it was evaluated using the test set. It can be seen that training without data augmentation indeed had better results reaching 96.43% of accuracy than the model trained with data augmentation which had 78.57% accuracy. At the same time, not using all the layers in the pre-trained model did improve the performance of the model as well, having an accuracy of 100% without augmenting the training data.

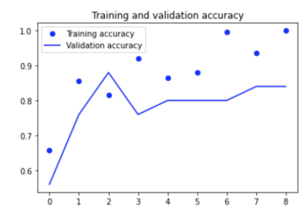

Fig. 33. Accuracy of the Model Trained Without Augmenting the Training Set (All Version)

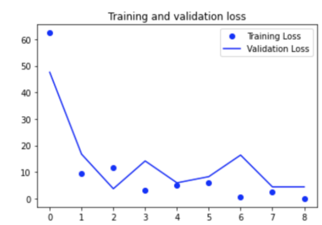

Fig. 34. Loss of the Model Trained Without Augmenting the Training Set (All Version)

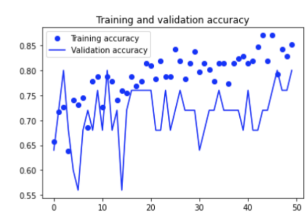

Fig. 35. Accuracy of the Model Trained With Augmented Training Set (All Version)

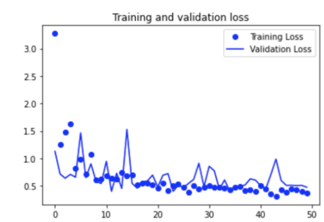

Fig. 36. Loss of the Model Trained With Augmented Training Set (All Version)

Next are the results of using the smart flooring dataset. For the smart flooring, it has two datasets, the All version and the Clean version. As seen in the graphs, the model of Figures 33 and 34 was able to obtain a validation accuracy of 88%. Meanwhile, the model of Figures 35 and 36 had attained 80%. Similar to the results of using the Kinect data, the model trained without augmenting the data had better performance based on the validation accuracies than the one trained with data augmentation.

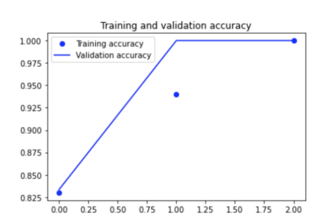

Fig. 37. Accuracy of the Model Trained Without Augmenting the Training Set (Clean Version)

Fig. 38. Loss of the Model Trained Without Augmenting the Training Set (Clean Version)

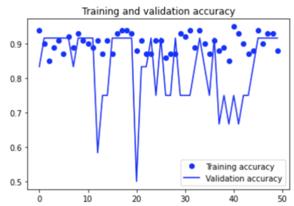

Fig. 39. Accuracy of the Model Trained With Augmented Training Set (Clean Version)

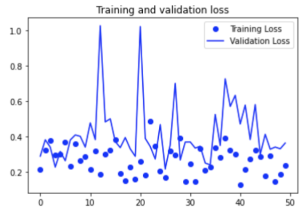

Fig. 40. Loss of the Model Trained With Augmented Training Set (Clean Version)

As for the Clean version dataset, similarly, the model trained without augmenting the data resulted in a higher accuracy. Another notable observation would be the performance of the model of Figures 39 and 40. It was inconsistent which could mean the model might be having a hard time learning. Highest validation accuracy achieved by the model of Figures 37 and 38 was 100% and 91.67% for the model of Figures 39 and 40.

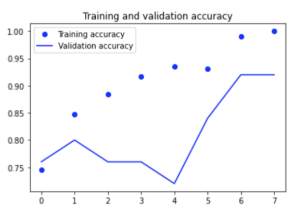

Fig. 41. Accuracy of the Model Trained Without Augmenting the Training Set (All Version)

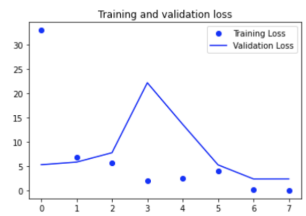

Fig. 42. Loss of the Model Trained Without Augmenting the Training Set (All Version)

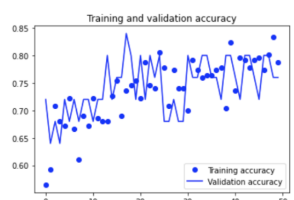

Fig. 43. Accuracy of the Model Trained With Augmented Training Set (All Version)

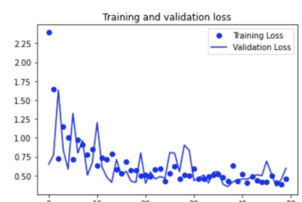

Fig. 44. Loss of the Model Trained With Augmented Training Set (All Version)

Not including all the layers in the pre-trained model was also implemented for the smart flooring data. Same with the results of not using all the layers for the Kinect data, the model had performed better when not all layers were used. Also, the model without data augmentation achieved higher accuracy as well. However, the loss values of the model with data augmentation are lower. This could mean that the model of Figures 41 and 42 was overfitted since its validation accuracy were lower than the training accuracy as well. The model was able to achieve 92% of validation accuracy for unaugmented training set and 84% for the augmented one.

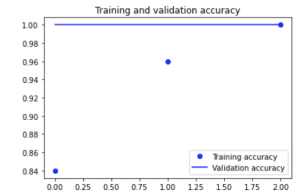

Fig. 45. Accuracy of the Model Trained Without Augmenting the Training Set (Clean Version)

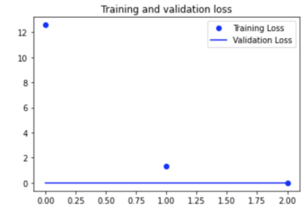

Fig. 46. Loss of the Model Trained Without Augmenting the Training Set (Clean Version)

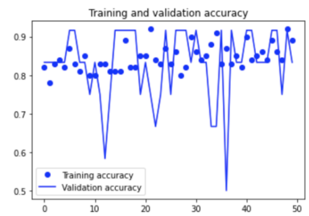

Fig. 47. Accuracy of the Model Trained With Augmented Training Set (Clean Version)

Fig. 48. Loss of the Model Trained With Augmented Training Set (Clean Version)

Same with the previous model, the clean version dataset produced better results than all version dataset. Without using all the layers in the pre-trained model for clean version dataset, it had achieved similar results with the model using all layers. The highest validation accuracies obtained are 100% for training without augmentation and 91.67% with augmentation. Consistently, the model without data augmentation still had higher results.

ALL Layers				
Train Dataset	Validation Dataset	Test Dataset	without data aug	with data aug
ALL	ALL	ALL	96.43%	96.43%
CLEAN	CLEAN	CLEAN	92.86%	92.86%
NOT ALL Layers				
ALL	ALL	ALL	92.86%	57.14%
CLEAN	CLEAN	CLEAN	92.86%	100.00%

Fig. 49. Accuracy of the Model Using Test Sets

It can be observed after testing the model with the test sets, all dataset was able to produce a better model in most cases, although during training time, the clean dataset had higher validation accuracies. This could indicate that the clean dataset had removed some important data from the training set which was needed to be able to recognize data from the test set. The all version dataset model had achieved 96.43% of accuracy, and the clean version dataset reached 92.86% of accuracy when all layers in the pre-trained model were included, for both with or without data augmentation. As for not using all the layers, the all dataset obtained 92.86% of accuracy when no data augmentation was applied to the training set and 100% for the clean dataset. In this case, the clean dataset performed better when data augmentation was used, unlike when using all the layers in the pre-trained model where all dataset had higher results. This means the number of layers is a great factor that affects the performance of the model.

D. Decision-level Fusion

The model that yielded the best accuracy result for the smart flooring is through using CNN with an accuracy of 96.43%. On the other hand, the highest accuracy using the Kinect model was using KNN with 92.59% accuracy. Using the accuracies mentioned, the weights were computed. A sample computation of weights is seen in Figure 50. The resulting weights produced a total of 52% for smart flooring and 48% for Kinect. Finally, the results of the decision-level fusion formula are compared to a threshold value of 50%. If the value is greater than 50, then the final output is abnormal but if the value is less than 50 then it corresponds to a normal gait. The results of these computations are compiled in Table I.

Kinect Weight = 92.59 / (92.59 + 96.43)
Mat Weight = 96.43 / (92.59 + 96.43).

Fig. 50. Sample computation of getting the weights

*48 * 1 + 52 * 0 = 48*
*48 * 1 + 52 * 1 = 100*
*48 * 0 + 52 * 1 = 52*
*48 * 0 + 52 * 0 = 0*

Fig. 51. Sample computation using decision-level fusion formula

Kinect Weight	Kinect Prediction	Smart Flooring Weight	Smart Flooring Prediciton	Result
48	1 (Abnormal)	52	0 (Normal)	$48 < 50 \therefore Normal$
48	1 (Abnormal)	52	1 (Abnormal)	$100 > 50 \therefore Abnormal$
48	0 (Normal)	52	1 (Abnormal)	$52 > 50 \therefore Abnormal$
48	0 (Normal)	52	0 (Normal)	$0 < 50 \therefore Normal$

TABLE I
DECISION-LEVEL RULE TABLE

Given that CNN yielded the highest accuracy for the smart flooring, it indicates that visualizing the smart flooring data helps determine the difference of a normal to an abnormal gait better. While for the Kinect, the KNN algorithm proved to give the best result, and could be interpreted that the numerical data points of the coordinates are important and having it visualized makes it more difficult for the algorithm to learn from. Given that the model of the smart flooring yielded a slightly higher result, in the case of contradicting results between the smart floor and the Kinect, the decision would end up adhering to the result of the smart floor due to its higher accuracy. Although the two sensors only had a difference of only 4% in terms of their weight, the smart flooring having a higher weight value shows that it proves to be a more useful tool in determining gait.

VIII. CONCLUSION AND FUTURE WORKS

In conclusion, the gait of a person can be captured through analyzing the Kinect data points or through the pressure points from the smart flooring. The results indicate that KNN and CNN have their own configurations that best fit the data.

KNN on Kinect showed that feature importance selection had little to no impact on the result of the machine learning algorithm due to the similarity in results while on the Smart Flooring it would be a mix of before feature importance selection and after feature importance selection that would produce the best results. The highest results from KNN on Kinect was 92.59% and it is higher than the highest result retrieved from the Smart Flooring which was only at 87.94%. This means that Kinect is able to differentiate gait abnormality better by almost 5%. However, Smart Flooring data which was at 87.94% came from a dataset where it was cleaned to remove all 0 values and it was the dataset that produced the best accuracy for the Smart Flooring.

CNN on the other hand was able to achieve 100% of accuracy for Kinect when the model didn't use all the layers in the pre-trained model and no data augmentation was done to the training set. When trained with augmented training data, the highest accuracy obtained was 82.14% without using all the layers in the pre-trained model. As for smart flooring data, 96.43% was reached for using all layers in the pre-trained model and the all version dataset as the training set with and without data augmentation, while 100% for the model using clean augmented dataset as training model and not all the layers in the pre-trained model. It can be deduced that both Kinect data and smart flooring data were represented-well visually since they have produced acceptable results. Another notable observation would be that without applying data augmentation to the training dataset, the models yielded higher results in most cases. However, the purpose of applying data augmentation is to reduce the chance of overfitting which may mean that the model was not good enough yet to handle cases not similar to the training set. It has to be more generalized. Also, not using all the layers in the pre-trained model, the model was able to have better performance. For the smart flooring data, it has been noticed that the all dataset had better results than the clean dataset.

Additionally K-Means proved to be useful in understanding the data better. The K-Means model with the lowest DBI values was able to cluster the smart flooring and Kinect data into 3 groups which were the abnormal, normal, and outlier data points. Therefore, this is a clear statement that it is able to distinguish abnormal and normal data. However, throughout all of the data, there were inaccuracies caused by overlapping data points. The abnormal points overlapping that overlapped with the normal points in the Kinect data were the ones acted out but for smart flooring, it was the real ones. This can mean that for smart flooring, the difference between the normal walk and the acted gait abnormality was more obvious because of the dragging of foot and smaller foot steps. It can also be deduced that the negative values removed in the clean smart flooring data may be playing an important role in differentiating normal from abnormal gait since the All dataset was clustered better than the Clean dataset.

Decision-level fusion aided in determining the gait abnormality of a person even further through the producing a confidence level. Gait abnormality can be better predicted through a smart flooring device having a base weight value of 52% as opposed to Kinect with only 48%. Although the smart flooring provides a better accuracy percentage, it has its cons and is more difficult to implement a smart flooring setup over a Kinect.

The proponents encourage future researchers to collect more data from people with normal and abnormal gait. Also, given that there are a lot of abnormal gait patterns, it would be better if the data collected would comprise various types of gait abnormalities so that the machine learning algorithms are given different examples of an abnormal gait and could classify more accurately. Furthermore, other state-of-the-art machine learning algorithms could be explored as well other than KNN, K-Means, and CNN. In addition to that, other fusion methods such as data level and feature level fusion could also be taken into consideration. This way, the results can be compared in order to discover which fusion method works best along with the model. Through this research, the potential of identifying the specific gait abnormalities can be explored by other researchers.

As for the hardware and environment setup, other ways to improve the portability of the smart setup can be explored. For example, they could lessen the wires connected to the smart floor so that it could easily be stored or brought. Moreover, instead of using just 1 Kinect camera placed on the side for a lateral view, the study could take one step ahead if 2 Kinect cameras were utilized; 1 placed in front and 1 on the side. This could yield more accurate coordinates of the joints as well as the length and angles. Also, it could capture some angles that are not fully covered when there is only 1 Kinect on the side like the swinging of the hips. Additionally, for an even more detailed and concise data of the footstep, it is encouraged to make the gap between each copper tape on the smart floor smaller. Lastly, a consultation or an interview with an orthopedic doctor or a licensed physical therapist before and after the data is collected might provide more insight about the study.

REFERENCES

[1] F. Horst, S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn, "Explaining the unique nature of individual gait patterns with deep learning," *Scientific reports*, vol. 9, no. 1, p. 2391, 2019.

[2] J. Fletcher, "What is abnormal gait?r," 2017. [Online]. Available: https://www.medicalnewstoday.com/articles/320481.php

[3] A. Mirelman, S. Shema, I. Maidan, and J. M. Hausdorff, "Chapter 7 - gait," in *Balance, Gait, and Falls*, ser. Handbook of Clinical Neurology, B. L. Day and S. R. Lord, Eds. Elsevier, 2018, vol. 159, pp. 119 – 134.

[4] J. Fletcher, "What is abnormal gait?" 2017. [Online]. Available: https://www.medicalnewstoday.com/articles/320481.php

[5] "Why is gait analysis important?" 2018. [Online]. Available: https://www.tekscan.com/blog/medical/why-gait-analysis-important

[6] "Importance of gait analysis and uses," 2016. [Online]. Available: https://www.mytechlogy.com/IT-blogs/2196/importance-of-gait-analysis-and-uses/#.XSdU7y2B1QK

[7] S. Chen, J. Lach, B. Lo, and G. Yang, "Toward pervasive gait analysis with wearable sensors: A systematic review," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 6, pp. 1521–1537, Nov 2016.

[8] M. Nieto-Hidalgo, F. J. Fernandez-Pastor, R. J. Valdivieso-Sarabia, J. Mora-Pascual, and J. M. García-Chamizo, "Gait analysis using computer vision based on cloud platform and mobile device," *Mobile Information Systems*, vol. 2018, pp. 1–10, 01 2018.

[9] M. Ye, C. Yang, V. Stankovic, L. Stankovic, and A. Kerr, "Gait analysis using a single depth camera," 12 2015.

[10] O. Oluwadare, "Gait analysis on a smart floor for health monitoring," 2015.

[11] S. V. K. Peddinti, "Smart shoe gait analysis and diagnosis: designing and prototyping of hardware and software," Ph.D. dissertation, 2018.

[12] Y. Watanabe, T. Hatanaka, T. Komuro, and M. Ishikawa, "Human gait estimation using a wearable camera," in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, Jan 2011, pp. 276–281.

[13] M. Hanlon and R. Anderson, "Real-time gait event detection using wearable sensors," *Gait & Posture*, vol. 30, no. 4, pp. 523 – 527, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0966636209003373

[14] S. Qiu, L. Liu, Z. Wang, S. Li, H. Zhao, J. Wang, J. Li, and K. Tang, "Body sensor network-based gait quality assessment for clinical decision-support via multi-sensor fusion," *IEEE Access*, vol. 7, pp. 59 884–59 894, 2019.

[15] "The difference between kinect v2 and v1," 2016. [Online]. Available: https://skarredghost.com/2016/12/02/the-difference-between-kinect-v2-and-v1/

[16] R. Rettner, "How many calories does walking burn? depends how tall you are," 2013. [Online]. Available: https://www.livescience.com/35159-walking-calories-burned-equation-101111.html#: :text=