Effectiveness of Classifiers to Identify Hand Gestures with Motion Capture Coordinate Markers

Arifur Rahman
School of Theoretical & Applied Science
Ramapo College of New Jersey
Mahwah, NJ
arahman1@ramapo.edu

Michael Whitlock
School of Theoretical & Applied Science
Ramapo College of New Jersey
Mahwah, NJ
mwhitlo1@ramapo.edu

Eman Abdelfattah
School of Theoretical & Applied Science
Ramapo College of New Jersey
Mahwah, NJ
eabdelfa@ramapo.edu

***Abstract*—In a world where technology is increasingly moving towards less wires, less devices, and more portability, the ability of a system to accurately interpret hand gestures is extremely valuable. The innovation of small, inexpensive cameras only makes this even more practical. The practicality of such a system rests on the assumption that hand gestures can be recognized from participants based solely on the shape of their hand. The dataset used for this research, while captured with node markers on a special glove, is formatted independent of a marker indexing system and could be applied in conjunction with a marker identification system for real world application of hand gesture classification. This paper presents an analysis and comparison of the effectiveness classifiers have when trying to determine hand gestures using motion capture coordinate markers provided by a Viacon camera. Stochastic Gradient Descent, Decision Trees, Logistic Regression, Random Forests, and Bagging are applied in this study. Overall, the Random Forests classifier performed the best with respect to performance measures such as recall, F$_1$ score, and precision.**

Index Terms – Viacon Camera, Multilayer Perceptron, Kernel Support Vector Machine, K-nearest Neighbor, Prazen, Geronimo-Hardin-Massopust, Stochastic Gradient Descent, Decision Trees, Logistic Regression, Random Forests, Bagging;

I. Introduction

The research performed examines the effectiveness of five machine learning models in conjunction with over 76,000 records of positional marker data. The classifier models used are Stochastic Gradient Descent, Decision Trees, Logistic Regression, Bagging, and Random Forests.

Stochastic Gradient Descent (SGD) is an iterative method for optimizing an objective function. As SGD is the most common optimization algorithm, it satisfactorily serves as a starting place for working with classification problems.

Decision Trees (DTs) are incrementally updated by splitting the dataset into smaller datasets, where the results are represented in the leaf nodes. Due to the large number of features, visualization of this model for this dataset is impractical.

Logistic regression (LR) predicts the outcome variable that is categorical from predictor variables that are continuous and/or categorical. This allows for modeling a nonlinear association in a linear manner. Since logistic regression is reputed as a powerful algorithm, it is selected as one of the machine learning models in this work.

Bagging improves the stability and accuracy of the analysis and results. Since Bagging classifier utilizes several different models internally, it has an ability to produce more accurate results and so it is selected for this experiment.

Random Forests (RFs) handle any missing values while maintaining the accuracy of a large proportion of data. Random Forests is chosen for this experiment to contrast against Bagging classifier and Decision Trees classifier, as they all utilize decision trees.

The dataset used contains multiple near identical records for any given gesture for any given user, therefore random sampling to achieve the separation of training from testing data does not apply. Instead, separation is performed on a per user basis at the recommendation of the dataset creators themselves.

From analysis of per user results, it becomes clear that more research into preprocessing of the data is necessary if the system is to be suitable for real world applications. Further research and development need to be done in the area of data imputation for missing values, as missing values are ubiquitous in this type of gesture data.

This paper is organized as follows: section II presents the related work. In section III, the dataset description is elaborated. Section IV contains experimental results and analysis. Finally, section V offers the conclusion.

II. Related Work

Mei *et al.* researched methods with higher discrimination machine learning classifiers applied to classifying hand postures [1]. They used a “slowest error growth” to discriminate at each boosting iteration among a set of stump classifiers. To reduce the number of features, they added an option to use mask images of the gestures instead of the full-color image, which also resulted in faster training and testing times. They found their methods to result in effectiveness and efficiency.

Kane *et al.* experimented with depth matrix and adaptive bayes classifiers to develop a practical framework for gesture recognition [2]. The model recognizes postures with a depth matrix and 1-nearest neighbor and the predictions were made with the bayes classifier in conjunction with adaptive

978-0-7381-4394-1/21//$31.00 ©2021 IEEE

windowing mechanism. Their accuracy was reported as 96.2% with mean accuracy of 95.2% in 2ms execution time.

Liu *et al.* focused on using conventional products to develop their fall detection system [3]. They preprocessed the data into monochromatic format for privacy reasons as well as to help downplay the role of upper limbs. They applied k-nearest neighbors model to classify using ratio and difference of the image as well as time difference. Using these combined, they had a prediction accuracy of 84.44% to detect falls and lying down.

Kelly *et al.* applied support vector machines in identifying hand postures [4]. Their unique approach, an Eigenspace Size Function, classified the postures on test users who were not in the training data. Their modified size function produced better results as far as performance over an unmodified one. Most importantly, their model performed competitively with other recognition systems.

Antwi-Afari *et al.* sought to classify awkward working postures of construction workers that lead to musculoskeletal disorders and ultimately occupational injuries [5]. They developed a non-invasive method using insole pressure measurements from footwear. They used artificial neural networks, k-nearest neighbors, decision trees, and support vector machines models. They found the SVMs classifier to perform the best with a 99.70% accuracy. They found that insole pressure was a valid method for classifying awkward working postures.

Zhao *et al.* researched and experimented with classifiers to identify different driving postures [6]. They applied multi-layer perceptron (MLP), intersection kernel support vector machine, k-nearest neighbor, and parzen. They found the best model to be feature extraction based on Geronmo-Hardin-Massopust (GHM) multiwavelet transform along with MLP. They noted that talking on a mobile device was the most difficult posture to classify.

Bush *et al.* created a system to control objects on a computer screen using live hand gesture capture [7]. The system hybridizes hand detection, prediction of hand position, and applies a deep learning algorithm. Specifically, they used a single shot multi box detector to detect the hand and a convolutional neural network for prediction. Their system did not require positional markers and used live images of hands. The authors approach was a great solution to any overhead received or gathered from processing multiple separate positional markers.

Silanon experimented with machine learning models to classify 21 hand postures used in Thai finger-spelling [8]. They applied a histogram of orientation gradient combined with adaptive boost. They selected the weakest classifier and constructed a strong classifier consisting of several of the weak ones. Different classifiers are selected for different postures based on the experimental results of false and true positives. They reported a 78% accuracy.

III. Dataset Description

The dataset comes from a study done by Andrew Gardner and Christian Duncan in 2014. There are 78,095 instances and 38 attributes. The dataset was created using a Vicon motion capture camera system. Twelve users performed five different hand postures with markers attached to a left-handed glove. Here, a pattern of markers on the back of the glove was used to establish a local coordinate system for the hand. Eleven other markers were attached to the thumb and fingers of the glove. Three markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. Two markers were attached to each finger with one above the fingernail and the other on the joint between the proximal and middle phalanx. There were eleven markers not part of the rigid pattern, which were left unlabeled. Their positions were not explicitly tracked.

Due to the resolution of the capture volume and self-occlusion and due to the orientation and configuration of the hand and fingers, many records have missing markers. Outlying markers are also possible due to the Vicon software's marker recording process and other objects in the capture volume. Therefore, the number of visible markers in a record varied.

The data is partially preprocessed. First, all markers are transformed to the local coordinate system of the record containing them. Second, each transformed marker with a norm greater than 200 millimeters is pruned. Any record that contained fewer than three markers is excluded. The processed data has at most twelve markers per record and at least three. Separation of training data from testing data was performed on a per user basis (of the dozen users who performed the gestures in the set) instead of via sampling. For a given record and user, it is likely that there exists a near duplicate record originating from the same user. Due to this caveat, it is recommended to evaluate classification algorithms on a leave-one-user-out basis so each user is iteratively left out from training and used as a test set. Then, one tests the generalization of the algorithm to new users. The 'user' attribute is provided to help with this strategy.

This dataset may be used for a variety of tasks. The most useful would be posture recognition using classification, which is used in this research. User identification is also possible. On the other hand, one can perform clustering, whether constrained or unconstrained in order to find marker distributions either as an attempt to predict marker identities or obtain statistical descriptions or visualizations of the postures.

To perform the classification, the target feature was hand gesture. Figure 1.1 shows the pie chart for the percentage of each hand gesture performed.

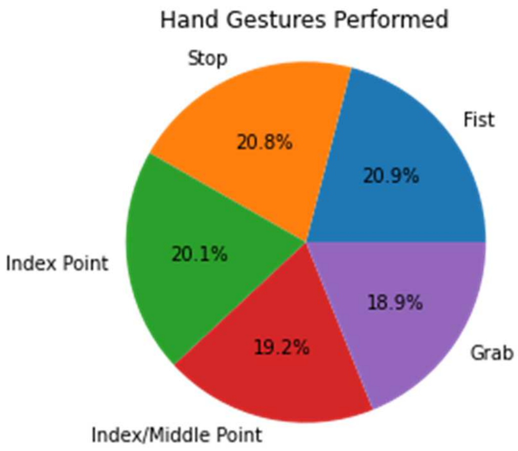

Figure 1.1 Pie chart showing the percentage of each hand gesture performed.

IV. EXPERIMENTAL RESULT AND ANALYSIS

In order to preprocess the dataset, a superfluous empty record had to be trimmed from the beginning. After assessing the number of records for each user, users 4 and 7 are excluded due to very low sample size as well as not performing all 5 of the gestures. Their other records have already been removed during preprocessing done before the release of the dataset. The dataset came ordered so it was shuffled randomly.

According to the dataset description, it contains multiple nearly identical records for any given gesture for any given user, so instead of splitting testing and training data with a random sampling, each user was isolated from the others to be the test data. Random sampling produced results of 1.0 for all models and constituted training with testing data.

The nature of the hand capture system creates a lot of missing values. Preprocessing that was done before release of the dataset trimmed all records with less than 3 motion capture markers and the markers were spread across the hand. Therefore, it is impossible to have data on all of them simultaneously from any given camera angle.

Since the capture system picked up marker coordinates without an indexing system, there is absolutely no guaranteed correlation between columns in the dataset. X0 of one record does not correspond to X0 of a different record. Therefore, standard imputing techniques do not apply. To impute missing values, all missing X values were replaced with the mean of all X values in the dataset, and likewise for Y and Z values. Once missing values were replaced, all values were normalized with a MinMaxScaler to between 0 and 1. While the method used for capturing directly led to all of the aforementioned difficulties in preprocessing, ultimately this is far more practical as many real-world applications would derive markers from raw image data of the gesture without a glove and would therefore provide no indexing of coordinates just as it is in this dataset.

Five different models were applied: Stochastic Gradient Descent (SGD), Decision Trees (DTs), Logistic Regression (LR), Bagging (B), and Random Forests (RFs). For each model, each user is used as test data against training data composed of the other users. Metrics performed on each model are Recall, F_1 score, Precision, training time, and testing time. Metrics are recorded for each user's testing and for each score a lowest, highest, and average are produced as shown in Figures 2.1, 2.2, 2.3, 2.4, and 2.5.

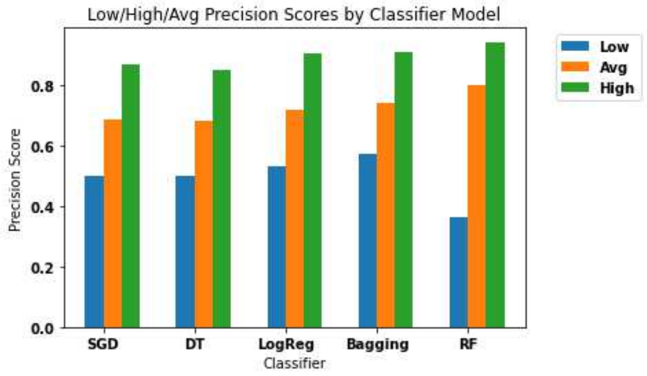

Figure 2.1 Lowest, highest, and average precision for each classifier model.

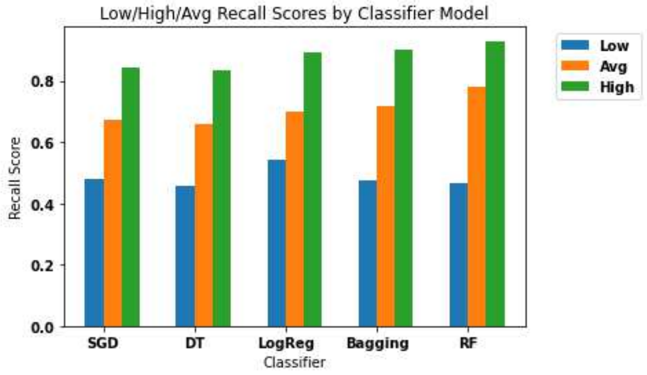

Figure 2.2 Lowest, highest, and average recall for each classifier model.

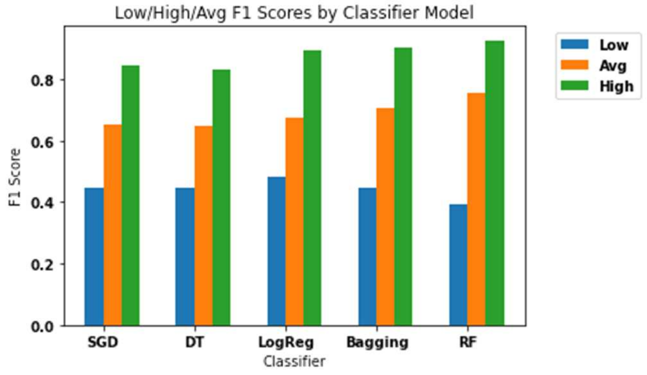

Figure 2.3 Lowest, highest, and average F_1 scores for each classifier model.

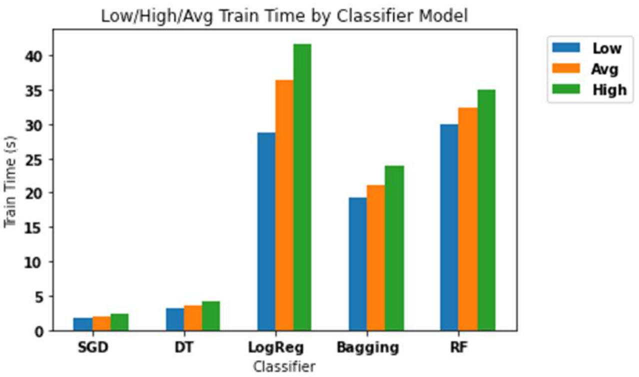

Figure 2.4 Lowest, highest, and average training time for each classifier model.

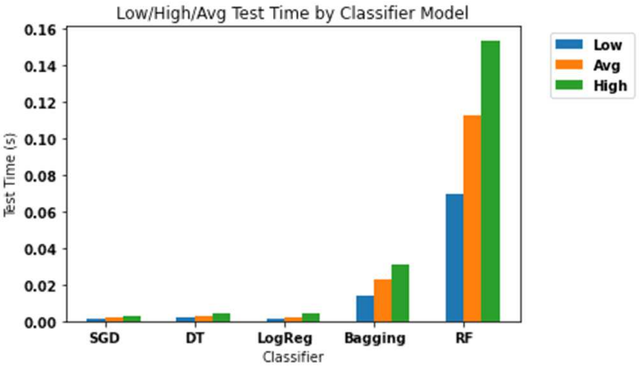

Figure 2.5 Lowest, highest, and average testing time for each classifier model.

An analysis of the confusion matrices show that misclassification instances are largely caused by similar hand gestures, such as fist vs. grab and index point vs middle point vs fist, and stop vs. fist. The confusion matrices for all used models are shown in Figures 3.1, 3.2, 3.3, 3.4, and 3.5.

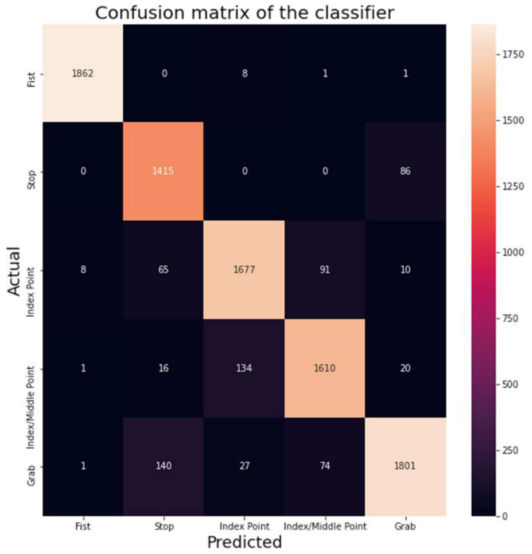

Figure 3.1 Confusion matrix of SGD with user 0 as the test user.

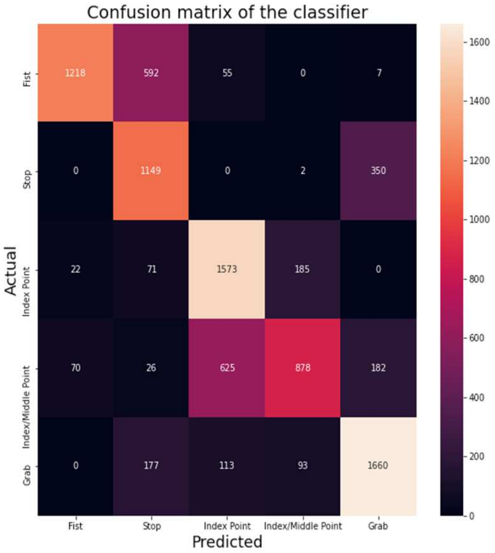

Figure 3.2 Confusion matrix of DTs with user 0 as the test user.

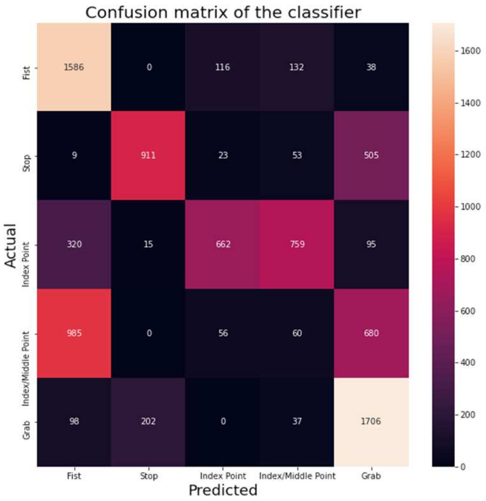

Figure 3.3 Confusion matrix of LR with user 0 as the test user.

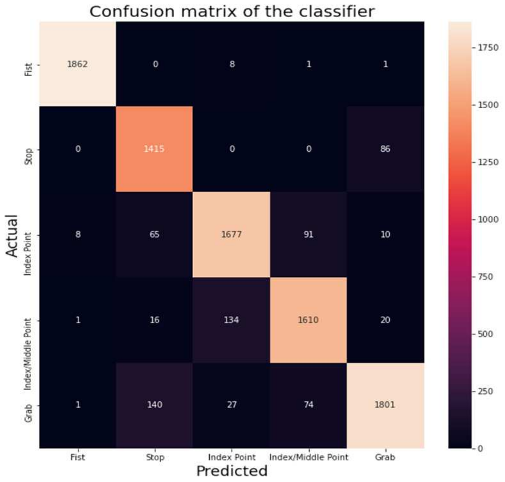

Figure 3.4 Confusion matrix of Bagging with user 0 as the test user.

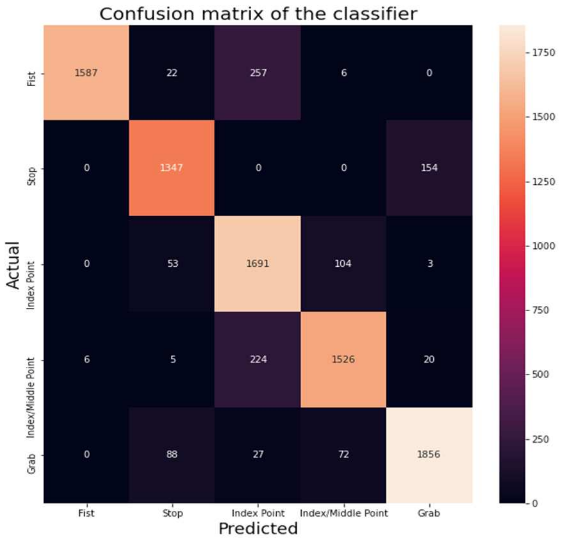

Figure 3.5 Confusion matrix of RFs with user 0 as the test user.

By analyzing the results per user, it becomes clear some users behaved as outliers and produced poorer results than the others. Specifically, user 5 performs poorly across all models, suggesting that more could be done to preprocess the data.

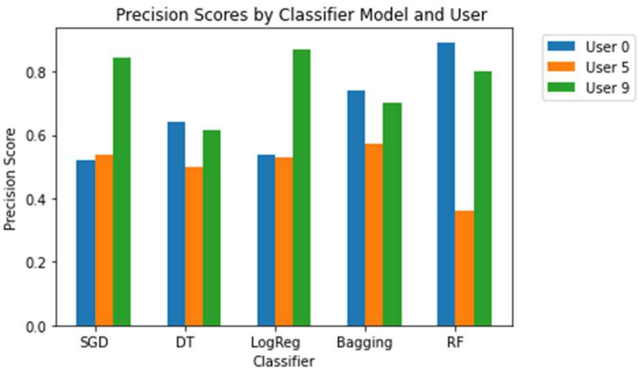

Figure 4.1 Precision for each classifier model for users 0, 5, and 9.

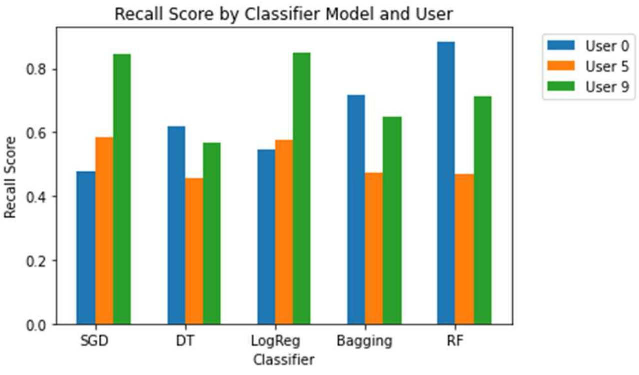

Figure 4.2 Recall for each classifier model for users 0, 5, and 9.

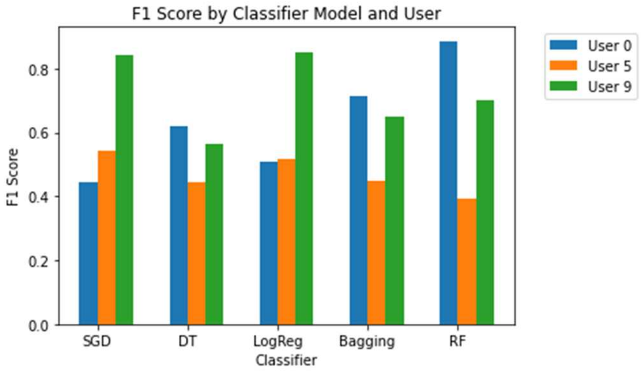

Figure 4.3 F_1 score for each classifier model for users 0, 5, and 9.

Figures 4.1, 4.2, and 4.3 show quite the disparity between results by user, confirming the need for more preprocessing.

Overall, the Random Forests classifier performs the best with respect to the performance measures of precision, recall, and F_1 score with an average precision of 0.801107, an average recall of 0.778556, and an average F_1 score of 0.754999. By contrast, the poorest performing model is unexpectedly Decision Trees, with an average precision of 0.682731, an average recall of 0.657941, and an average F_1 score of 0.646914.

V. Conclusion and Future Work

The used models have a comparable performance, but all models fail to work well for all users. Testing time is negligible for all classifiers except RF.

As mentioned in Section IV regarding the poor performance of user 5, a method for making the data more universal across different users is needed. One possible method would involve establishing a mean coordinate per gesture per user, and adjusting all coordinates for that user's gesture around that mean as (0,0,0). This may serve to improve performance for outlier users such as user 5.

User 5 could possibly be slightly out of the camera's view, which in turn gave us the outlying results. User 5's gestures were most likely out of frame or favoring one side, which the camera could not get a sufficient image of.

Additionally, other methods of data imputing could be attempted to replace the missing values. Instead of a generalized mean, a mean could be established per gesture per user and applied. This would not bleed any information into the testing group as all calculations would be per user.

VI. References

[1] K. Mei *et al.*, "Training more discriminative multi-class classifiers for hand detection," *Pattern Recognition,* vol. 48, no. 3, pp. 785-797, 2015.

[2] L. Kane *et al.*, "Depth matrix and adaptive Bayes classifier based dynamic hand gesture recognition," *Pattern Recognition Letters*, vol. 120, pp. 24-30, 2019.

[3] Chien-Liang Liu, Chia-Hoang Lee, and Ping-Min Lin, “A fall detection system using k-nearest neighbor classifier,” *Expert Systems with Applications*, vol. 37, no. 10, pp. 7174-7181, 2010.

[4] Daniel Kelly, John McDonald, and Charles Markham, “A person independent system for recognition of hand postures used in sign language,” *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1359-1368, 2010.

[5] Maxwell Fordjour Antwi-Afaria, Heng Lib, Yantao Yua, and Liulin Konga, “Wearable insole pressure system for automated detection and classification of awkward working postures in construction workers,” *Automation in Construction*, vol. 96, pp. 433-441, 2018.

[6] Chihang Zhaoa, Yongsheng Gaob, Jie Hea, and Jie Liana, “Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1677-1686, 2012.

[7] Idoko John Bush, Rahib Abiyev, and Murat Arslan, “Impact of machine learning techniques on hand gesture recognition,” *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 3, pp. 4241-4252, 2019.

[8] K. Silanon, “Thai Finger-Spelling Recognition Using a Cascaded Classifier Based on Histogram of Orientation Gradient Features”, *Computational Intelligence & Neuroscience*, pp. 1-11, 2017.

[9] A. Gardner, R. R. Selmic, C. A. Duncan, and J. Kanno, “Motion Capture Hand Postures Data Set,” UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Motion Capture Hand Postures. [Accessed 11 December 2020].