Off-line Inertial-Sensor Based Hand Gesture Recognition and Evaluation

Mirela Kundid Vasić[(1)], Tamara Grujić[(2)], Ivo Stančić[(2)], Josip Musić[(2)], Mirjana Bonković[(2)],

[(1)]Faculty of Mechanical Engineering, Computing and Electrical Engineering, University of Mostar, Mostar, B&H
(e-mail: mirela.kundid.vasic@fsre.sum.ba)
[(2)]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture - FESB, Department of Electronics and Computing, University of Split, Split, Croatia
(e-mail: tamara.grujic@fesb.hr, ivo.stancic@fesb.hr, josip.music@fesb.hr, mirjana.bonkovic@fesb.hr)

***Abstract*— Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures with computer programs and many different algorithms. Any bodily motion or state that most commonly originates from the face and/or hand can be interpreted as a human gesture. Most of the research today focuses on emotion detection and recognition of hand gestures using cameras and computer vision algorithms. Gesture recognition can be seen as the way computers begin to understand human body language. There are many different areas this topic of computer science can be applied to; the main field is human-computer interaction interfaces (HCI). Today the main interaction tools between computers and humans are still keyboard and mouse. Gesture recognition can be used as a tool for communication with the machine and interact without any mechanical device such as keyboard or mouse. In this paper, we present the results of a comparison of five different machine learning classifiers in the task of human hand gestures recognition. Gestures were recorded by using inertial sensors, gyroscopes, and accelerometers placed at the wrist and index finger. One thousand and eight hundred (1800) hand gestures were recorded and labelled. Six important features were defined, for the identification of nine different hand gestures, using five different machine learning classifiers: Logistic Regression, Random Forests, Support Vector Machine (SVM) with linear kernel, Naïve Bayes classifier, and Stochastic Gradient Descent.**

***Keywords*—**hand gestures, inertial sensors, machine learning algorithms, off-line classification, evaluation**.**

I. INTRODUCTION

Nowdays, in a world where computers become more and more pervasive in culture, the need for efficient human-computer interaction is increasing at a rapid pace. The most commonly used way of human-computer interaction (HCI) is a graphical user interface (GUI) which requires the use of additional devices, e.g. mouse, keyboard, etc. Researchers in academia and industry are increasingly looking for ways to make human-computer interaction easier, safer, and more efficient. Consequently, new interaction styles have been explored. One of them is hand gesture-based interaction, which allows users a more natural way to communicate, without any extra devices, which is much simpler and intuitive than using graphical interfaces or text input. For example, such type of interaction is used in applications like controlling smart interactive television [1] or enabling a hand as a 3D mouse.

Recognized gestures also can be used for controlling a robot [2] or conveying meaningful information. It can be very useful in particular situations, e.g. with robots designed to assist disabled people and helping them with personal and professional tasks on a daily basis, or for real-time mobile robot control [2].

The key problem in hand gesture-based interaction is how to make gestures easily understandable and accurately interpretable to computers. Thus, hand gesture recognition is an area of active research in the field of computer vision and machine learning. Accordingly, different approaches have been considered and all of them can be mainly divided into two groups: vision-based [3], [4] and inertial sensors-based methods, which are reliable, accurate, and robust, for work in real-time [2], [5]. These methods

involve the use of inertial sensors placed on the user's body.

In gesture recognition, two main popular approaches are available: body-worn systems that track user gesture and motion using body worn sensors attached to the human body, or external systems based on external sensors. Sensors like Microsoft Kinect allow human identification and tracking in 3D space using RGB-D sensors which combine RGB colour information with per-pixel depth information, to get information about objects in 3D space. Many researchers have reported successful usage of hand gesture recognition tasks for control of a variety of robots such as mobile robots [2], [6], or humanoid robots [7]. Most of the research is based on inertial sensors that use machine learning methods for gesture identification. The gesture identification task consists of several steps, including the recording of hand gestures and their classification, which is called hand gesture modelling. The process of hand gesture modelling is the main topic of this research and implies the selection of the right identification units of the movement also called features.

II. Features Definition and Selection

Feature selection is the most important task in classification, and the main goal is to determine the right set of identification units for the particular classification task, or in this case, movement, to most accurately determine the difference between movements in the particular set. The movements that we are trying to model are shown and described in detail in [2]. Nine gestures are simple to perform since they consist of few elementary hand motions and are devised in a manner that it is possible to extract at least one significant unique feature from each of the nine described gestures.

The quality of the features is the most important factor in the classification task. The hand gesture modelling includes the identification process of the features that are most discriminative using the subset of initially proposed features.Features used in this approach rely on data obtained from a combination of gyroscopes and accelerometers signals. The inertial sensors (accelerometers and gyroscopes) are connected to the wrist and index finger to obtain the empirical data needed to recognize the motion. Wearable sensors should be small and lightweight, in order to be fastened to the human body without compromising the user's comfort and allowing her/him to perform the movement under unrestrained conditions as much as possible.

A pattern recognition machine does not perform classification tasks working directly on the raw sensor data. Usually, before the classification, data representation is built in terms of feature variables [8].

The features used in this research are listed in Table I, [2]. The suggested features are extracted from gyroscopes and accelerometers and are hand-labelled for certain hand motions/gestures. The first feature is gesture duration, which in some cases can't be a discriminating feature because it could be the same duration for different gestures. By further analysis, additional features are obtained, like the second feature that contains local extremes of gyroscope differential data (number of extremes), while the gyroscope axis ratio is the third feature implemented. The fourth and fifth features are derived from accelerometers data and that is accelerometer axis ratio, which represents absolute acceleration, and movement energy. The last feature is the magnitude of the first significant extreme in the gyroscope data.

TABLE I
FEATURES USED FOR OFF-LINE EVALUATION WITH SELECTED CLASSIFIERS [2]

Feature name	Feature description	Sensors used *
Gesture duration	Gesture duration in ms	G1, G2
Number of extremes	Number of extremes from differential gyroscope data (DGD)	G1, G2
Gyroscope axis ratio	Mean ratio of axis of DGD, detects direction of motion	G1, G2
Accelerometer ratio	Mean ratio A1 axis, detects hand orientation	A1
Movement energy	Integrates absolute A1 and A2 magnitude over the duration of the whole gesture	A1, A2
First rotation direction (flexion or extension)	Magnitude of the first large DGD peak, detects hand rotation direction	G1, G2

*As in Fig. 2: G1 is the wrist gyroscope, G2 is the index finger gyroscope, A1 is the wrist accelerometer, A2 is the index finger accelerometer

Experimental setup and generation of the model for all nine gestures included twenty participants, each of them had ten measurements of all nine performed gestures, to generate the reference point for every gesture. Therefore, the proposed model included 1800 labelled hand gestures to form the

baseline for the gestures included in process of classification. The hand gestures labelled data is used for the model generation using five different classifiers that will be explained in Section III.

III. Classification Task

The classifiers used in this research are the Naïve Bayes classifier, Support Vector Machine (SVM) with linear kernel, Logistic Regression, Random Forests, and Stochastic Gradient Descent. The classification task for each of five different classifiers is the same and consists of two phases – the training phase and the classification phase. Please note that the off-line classification approach is used, and not a real-time one. The data was collected during the feature extraction phase and then used to determine the best classifier based on the scoring classifier phase. The scoring phase first divided the dataset into training and testing sets in a 50:50 percent ratio. Then it trained the selected classifier and tested the classifier on the test set. For each of the classifiers the precision, recall, and F-Score have been calculated, to show which classifier will achieve the best score. Classification of data collected from the hand gesture collection phase has been done using SkLearn library with Python programming language.

A. Naïve Bayes Classifier

The Naive Bayes (NB) [9] classifier method consists of two phases: the training phase and the classification phase. The testing has been done on the test set of data from a total of nine hand gestures by the Naive Bayes classifier. Naive Bayes classifier gives a statistical dimension to the made conclusions. Membership to each cluster (class) is determined by the distribution of probabilities. Therefore, optimal classification can be determined by taking into consideration the distribution of probabilities to which each vector belongs (aligning each feature in each group). Data is presented as an *n*-dimensional vector; the classifier's task is to predict a group of testing data based on equation (1).

$$\underset{V_j \in V}{argmax} p(v_j | a_1, a_2, a_3, \dots, a_n) \quad (1)$$

If Bayesian theorem is applied to Equation (1), Equation (2) is obtained.

$$V_{NB} = \underset{V_j \in V}{argmax} p(v_j) \prod p(a_i | v_j) \quad (2)$$

Set V_{NB} denotes the classified instance and its probability of belonging into a certain class, in this case, one of nine hand gestures.

B. Support Vector Machine

Support Vector Machine (SVM) [10] represents a set of related supervised learning methods (supervised learning) that are used for classification and regression. The SVM is a binary classifier i.e. probabilistically classifies into two categories. The SVM classification is based on a division of all points in space into two categories according to the margin between support vectors, and the algorithm searches for the largest gap between the two categories.

This procedure is referred to as the so-called linear classification, however, to perform the classification of several categories, we perform a kernel trick that implicitly maps the inputs into the multidimensional space. The trick avoids explicit mapping, which is necessary to get a linear learning algorithm to be trained with a nonlinear function. Classifier training creates boundary decisions separate margin that must be maximum, and the input data set, we can get a linear distribution of the two classes. Computing the SVM classifier amounts to minimizing an expression of the form denoted in equation (3).

$$\left[\frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i (w * x_i + b)\right)\right] + \Lambda ||w||^2 \quad (3)$$

The y_i are labels for each data sample x_i, w is normal vector that separates the data into two planes and b is the margin between hyperplane and classified data. The parameter Λ denotes the trade-off between increasing the margin size and ensuring that x_i is inside the right plane.

C. Logistic Regression

Logistic regression (LR) [11] is a regression model where the dependent variable is

categorical. The binary logistic model is used to estimate the probability of binary response, based on one or more predictor variables (features). LR measures the relationship between the categorical dependent variable and one or more independent variables using a logistic function. Conditional distribution can be Bernoulli or Gaussian because the outcome of the event can be binary (the dependent variable can have only two values). Mathematically, LR is the task of estimating log odds of a certain event, and it estimates multiple linear regression functions as denoted in equation (4).

$$logreg(p) = log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 * x_{i1} + \beta_2 * x_{i2} + \cdots + \beta_p * x_{ip} \quad (4)$$

D. Random Forests Algorithm

Random forest classifier (RFC) [12] algorithm is a notion of the general technique of random decision forests that is a learning method for classification, regression, and other tasks. Random forest is a collection of decision trees whose results are aggregated into one final result. The algorithm operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Given a training set $X = [x_1, x_2, \dots, x_n]$ with corresponding labels $Y = [y_1, y_2, \dots, y_n]$, the algorithm selects a random sample with replacement of the training set and fits trees to these samples and does that repeatedly B times. After training, unseen samples x' can be classified by averaging predictions on individual trees (regression), as shown in equation (5), or by taking the majority vote in decision trees.

$$f' = \frac{1}{b}\sum_{b=1}^{B} f'_b(x') \quad (5)$$

E. Stochastic Gradient Descent

Stochastic gradient descent (SGD) [13] is a gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. In both gradient descent and stochastic gradient descent, a set of parameters is updated, in an iterative manner, to minimize an error function. SGD is one of the fastest training algorithms. SGD is popular within training wide range of models in machine learning and is a de-facto standard for training artificial neural networks. This problem is considered as the problem of minimizing an objective function as denoted in equation (6). The parameter w is to be estimated, whilst Q_i is associated with observations in the dataset.

$$Q(w) = \sum_{b=1}^{B} Q_i(w) \quad (6)$$

IV. Classification Results, Comparison, and Discussion

The comparison of the different classifiers has been done using the confusion matrices for each of the defined classifiers. The main concepts of confusion matrices are false positive observations (hereinafter denoted as FP), false negatives (FN), true positives (TP), and true negatives (TN). The evaluation measures for the scoring of the particular classifier were Precision, Recall, and F1 – score.

The results of the classification are generated by Python programming language using SkLearn library to calculate the classes based on generated models. Five different classifiers were compared, based on precision, recall, and F1 score, as shown in Table II.

TABLE II
PRECISION, RECALL AND F1 SCORE MEASURES FOR CLASSIFIERS COMPARISON

	Precision	Recall	F1-score
Random forest classifier (RFC)	0.974	0.973	0.973
Logistic regression (LR)	0.973	0.972	0.972
Linear SVM	0.956	0.955	0.955
Naïve Bayes (Gaussian)	0.936	0.934	0.934
Stochastic gradient descent (SGD)	0.288	0.412	0.290

When the problem of classification involves the search for the positive class samples and they are very rare compared to the negative classes, the precision and recall approach is used. This method for evaluation of classifiers is more useful in "needle-in-haystack" type problems where the positive class is more "interesting" then the negative class. When

it is needed to emphasize negative class, the Receiver Operating Characteristic (ROC) plot is used. ROC curves represent a graphical plot of True Positive Rate (TPR) as the function of False Positive Rate (FPR). ROC curves for all five classifiers are shown in Fig. 1 - Fig. 5. Figures show ROC curves for all nine classes (nine hand gestures).

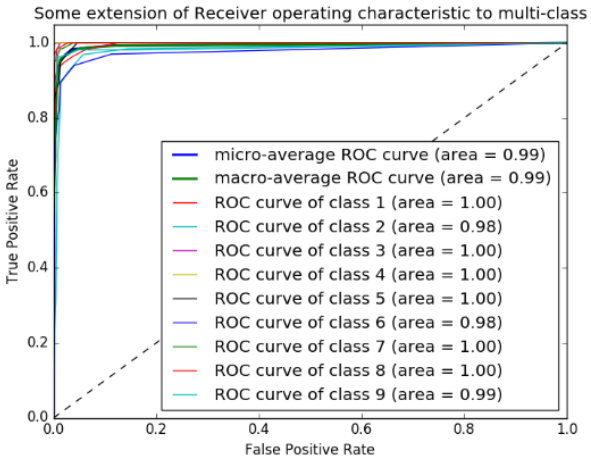

Fig. 1 ROC plot for Random forests classifier (micro averaged 99%)

The third classifier is SVM, whose advantages are the high accuracy, nice theoretical guarantees regarding overfitting, and with an appropriate kernel, they can work well even if data is not linearly separable in the base feature space. SVM is especially popular in text classification problems where very high-dimensional spaces are the norm (sparse data). In this case, where the data is clearly linearly separable, SVM with the right parameters can achieve the best results, but the main disadvantage is the speed due to which it is impossible to use this classifier in real-time scenarios.

One of the simplest classifiers is the Naïve Bayes algorithm, which is also, as Logistic regression, probabilistically oriented. Naïve Bayes classifier will converge quicker than discriminative models like Logistic regression, so less training data is needed. However, this algorithm is less accurate, as shown in PR values and ROC plots.

The last classifier that is tested in this comparison, Stochastic Gradient Descent (SGD), is the worst based on used performance parameters. SGD is one of the on-line algorithms that uses a combination of linear functions to solve the minimization problem. Despite the obvious lack of precision, recall, and F1-score in this off-line example, the main advantage of SGD is the light computational cost that is suitable for on-line approaches.

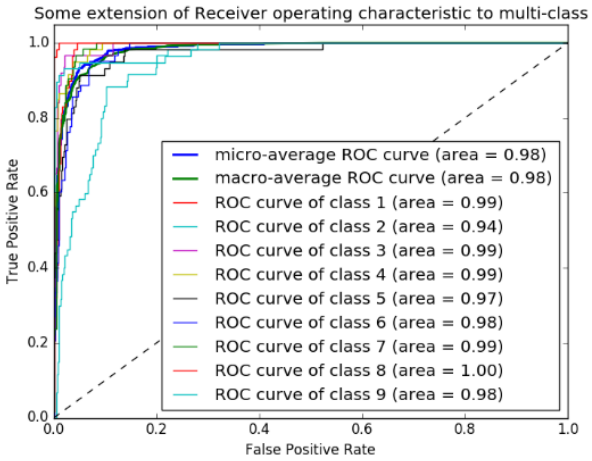

Fig. 2 ROC plot for Logistic regression classifier (micro averaged 98%)

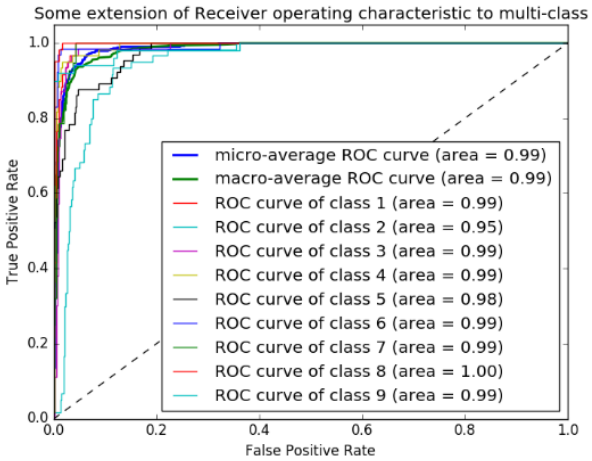

Fig. 3 ROC plot for Support Vector Machine classifier (micro averaged 99%)

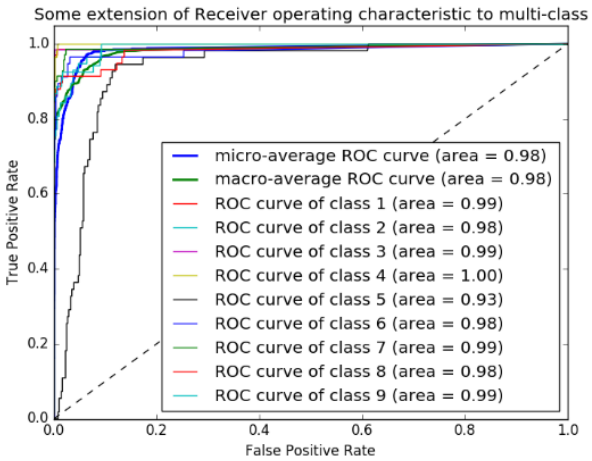

Fig. 4 ROC plot for Gaussian Naïve Bayes classifier (micro averaged 98%)

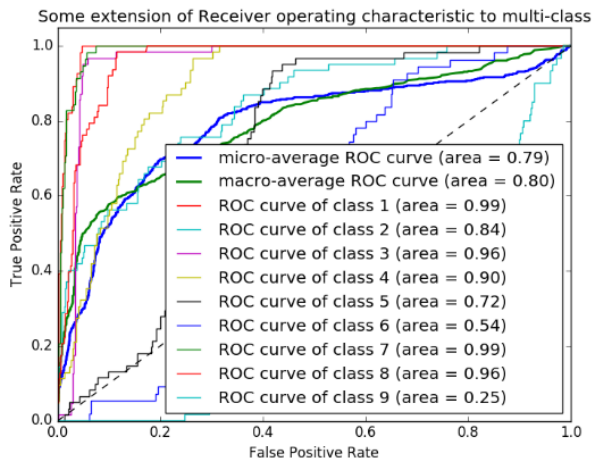

Fig. 5 ROC plot for Stochastic gradient descent classifier (micro averaged 79%)

V. Conclusion

The main goal of this research was to test the performance of five different machine learning algorithms to get a robust method for the classification of human hand gestures. All tested methods depend on the data that is presented as training data set, therefore the features that are used are a very important factor in a classification task. The data and features provided in [2] are highly discriminative and, as obtained results show, provide good classification results with tested machine learning (ML) algorithms.

All of the classifiers, except the SGD algorithm, showed very high F1-score and Area Under Curve (AUC) in ROC plots. The best classifier for the real-time classification scenario might be the Logistic Regression classifier, which showed almost identical results on testing data as Random forests classifiers. However, Random forests classifiers are not suitable for real-time classifications.

Better data often outperforms better algorithms, and thus the design of high-quality features is of great importance. With a sufficiently large dataset and high-quality features, different classification algorithms can perform well in terms of classification performances. Therefore, the choice of which algorithm to use should be based on performance speed or ease of use. In the case of Logistic Regression, different methods for model regularization are available, and the features correlation issue should not be a problem. Therefore, there are many arguments why and how this classifier can be successfully used in an on-line scenario of hand gesture recognition based on inertial sensors.

References

[1] Vishwakarma D. K. and Kapoor R, "An Efficient Interpretation of Hand Gestures to Control Smart Interactive Television," *International Journal of Computational Vision and Robotics*, Vol. 7(4), pp. 454 – 471, 2017.

[2] Stančić I., Musić J, and Grujić T., "Gesture Recognition System for Real-Time Mobile Robot Control Based on Inertial Sensors and Motion Strings," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 33-48, 2017.

[3] Kundid Vasić M., Galić I., and Vasić D., "Human Action Identification and Search in Video Files," in Proc. of 57th International Symposium on Electronics in Marine - ELMAR, IEEE, 2015.

[4] Choondal J. J. and Sharavanabhavan C., "Design and Implementation of a Natural User Interface Using Hand Gesture Recognition Method," *International Journal of Innovative Technology and Exploring Engineering*, vol. 10(3), pp. 249-254, 2013.

[5] Xu R., Zhou S., and Li W. J., "MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition," *IEEE Sensors Journal*, vol. 12(5), pp. 1166-1173, 2012.

[6] Filaretov V., Yukhimetsa D., and Mursalimov E., "The Universal Onboard Information-Control System for Mobile Robots," in Proc. of the *25th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM*, Vienna, 2014.

[7] Riek L, Rabinowitch T., Bremner P., Pipe A., and Fraser M., "Cooperative Gestures: Effective Signaling for Humanoid Robots," in *Proc. of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, 2010.

[8] Mannini A. and Sabatini A. M., "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers," *Sensors*, vol. 10(2), pp. 1154-75, 2010.

[9] Pardos Z. and Heffernan N., "Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing," in *Proc. of the International Conference UMAP*, 2010.

[10] Cortes C. and Vapnik V., "Support - Vector Networks," *Machine Learning*, vol. 3, pp. 273–297, 1995.

[11] McCullagh P. and Nelder J., *Generalized Linear Models*, CRC press, 1989.

[12] Wai M. M. N., "Classification Based Automatic Information Extraction," in *ACELAE'11 Proc. of the 10th WSEAS International Conference on Communications, Electrical & Computer Engineering*, 2011.

[13] Chen J. and Melo G. D., "Semantic Information Extraction for Improved Word Embeddings," in *Proc. of NAACL-HLT*, Denver, Colorado, 2015.