**ABSTRACT**

# Project description

Hand Gesture Recognition is a dynamic area of research. For different application, different gesture recognition techniques are used. In this project hand gesture recognition is implemented for digit recognition. A model is developed from android accelerometer data collected using an android app to recognize numerical digits by hand gesture.

# Data collection methodology

The numbers are drawn in air with the mobile phone in hand, its screen facing the ceiling. The mobile phone is used to collect the data and recognize the digits. The data is collected, using the ‘IMU+GPS+Stream’ app, which provides the user with the accelerometer data in .csv format. The gestures (with the phone in hand) were made in the X-Y plane (plane parallel to the ground). Data from 5 different subjects was collected at a frequency of 100 Hz (data transmission frequency).

The raw data was first normalized to -1 to 1 and various features were extracted. These features include skewness, kurtosis, etc.

# Feature Engineering

The collected DataStream for each digit was stored in .csv format. The rows represent the accelerations at each point of time and the columns represent the axis (x- and y-axis) about which the acceleration is being measured. We have taken about 150 data for each digit (0-9). From each DataStream, we have extracted 24 features about each axis, (48 in total) which are the representatives of each DataStream. So, the DataStream of each digit was converted into a row matrix of size 49, the first 48 columns being the features and the last one being the digit (0-9), which the data corresponds to. Our final feature matrix was of size 1636x48. On this feature matrix, we trained various machine learning models. The features we have used are explained below (all the features are of *float* datatype) –

1. **Statistical description:** 
   1. *Maximum and Minimum Value*: Maximum positive and negative amplitudes: vertices above and below the floating mean, respectively.
   2. *Standard Deviation:*
   3. *Skewness:* A measure of asymmetry of the distribution.
   4. *Kurtosis:* a measure of flatness of the distribution.
2. **Application of derivatives:**
   1. *1st derivative Mean and Max:* Mean and maximum value of the first derivative of the signal.
   2. *2nd derivative Mean and Max:* Mean and maximum value of the second derivative of the signal.
3. **Hjorth parameters:** The Hjorth parameters are simple measures of signal complexity. These measures are clinically useful tools for the quantitative description of a signal.
   1. *Activity:* It is the standard deviation of the signal.
   2. *Mobility:*
   3. *Complexity:*
4. **Frequency analysis:** The basic transform used in data processing is the Fourier transform (FT) that converts data signals from the time domain to the frequency domain. Nowadays, the discrete Fourier transform (DFT) is used to calculate the FT. An efficient algorithm to compute the DFT is the fast Fourier transform (FFT). FFT-based features calculated from each DataStream are mentioned below –

|  |  |
| --- | --- |
| Feature name | The mean value of a function over the following frequency bands: |
| FFT Delta | 0.1-3 Hz |
| FFT Theta | 3-7 Hz |
| FFT Alpha | 7-12 Hz |
| FFT Beta | 12-30 Hz |

1. **Wavelet Transform:** The wavelet transform (WT) is another option for time-frequency analysis. It is capable of distinguishing very small and delicate differences between time-varying signals even from short segments, and may describe highly irregular and non-stationary signals. Unlike the FFT-based features, WT-based methods can better localize the signal components in time-frequency space. WT-based features calculated (both detailed and approximate) – Mean, Standard deviation, Energy and Entropy.

# Model Training

1. **Decision Tree**

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables. Used K – Fold cross validation (K=10) with setting the impurity parameter i.e Gini Impurity to 10-5 and achieved 60% average accuracy on test dataset.

1. **K- nearest Neighbour**

KNN can be used for both classification and regression predictive problems. Used K – Fold cross validation (K=10) with setting the N parameter to 15 and achieved 66% average accuracy on test dataset.

1. **Logistic Regression**

Logistic Regression is a classification algorithm. Logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable (log of odds = log(p/1-p)).Used K – Fold cross validation (K=10) with setting the C(Regularization) parameter to 10000 and achieved 77% average accuracy on test dataset.

1. **SVM (Support Vector Machine**

SVM is a supervised machine learning algorithm which can be used for both classification and regression challenges. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well. Used K – Fold cross validation (K=10) with Polynomial Kernel and achieved 73% average accuracy on test dataset.

1. **ANN (Artificial Neural Network)**

Working of ANN takes its roots from the neural network residing in human brain. ANN operates on something referred to as Hidden State and these hidden states are similar to neurons. Each of these hidden state is a transient form which has s probabilistic behaviour. A grid of such hidden state act as a bridge between the input and the output. Used K – Fold cross validation (K=10) with setting the hidden\_layer\_sizes(46,42), tanh function and Alpha(1e-4) parameters and achieved 66% accuracy on test dataset.

1. **Extreme Gradient Boosting(XGB)**

XGBoost is an optimized distributed gradient boosting library. XGBoost belongs to a family of boosting algorithms that convert weak learners into strong learners.In simple words, gradient descent tries to optimize the loss function by tuning different values of coefficients to minimize the error. Used K – Fold cross validation (K=10) with parameter tuning learning\_rate(0.1), n\_estimators(500), max\_depth(3), objective (multi:softmax) and achieved 80% accuracy on test dataset.

From the above model performance, we can see that our **xgboost** model is performing best in our dataset and it gives more robust results as well. So, we finally used that model for real time prediction.

# Real time prediction:

To predict gestures in real time, we established a Wi-Fi connection between laptop and mobile device. Mobile transfers real time acceleration data to laptop. To extract the features of our interest, we stored a DataStream of length 256 in a numpy array and then applied the feature extraction functions on it. After creating the row matrix (1x48) of features, we fed it to our model to predict which digit-writing gesture is performed.

# **Future Projections**:

* Automatic conversion to text while writing
* Advanced Gaming
* Could be used as a control medium of home appliances for physically impaired people.
* Communication medium for Deaf and Dumb people through Hand Gestures.