Digital Pen: Writing gesture recognition using accelerometer.

Sarang Pande
(110560623)
Stony Brook University
spande@cs.stonybrook.edu

Amos Paul
(110898757)
Stony Brook University
apaul@cs.stonybrook.edu

INTRODUCTION

For many people, taking notes is an important task and so is having a digitized version of them. In this project, we aim to create an accelerometer based device which can be attached to any pen/pencil to record the text that was written. The device can have multiple applications when paired with different software e.g. It can be used of taking notes in class and instantly converting them into digital format or a Calendar or To-do app which is linked with the device and when used to write specific content will automatically create an event in your calendar.

We created a prototype for the device from scratch right from 3d design to 3-D printing along with using accelerometer ADLX335 and Arduino UNO board. We used GRT with processing for training our machine learning model.

BACKGROUND/ RELATED WORK

Lot of similar devices are available in market which transcribes what a user writes but these devices are expensive and requires special kind of paper to record writing. So, the idea was can we create a device which could perform similar task but cost less, work on normal paper and can be created by any person with little knowledge of electronic and machine learning.

We drew some of the inspiration for the product from the following papers [3][4], where the author has used complex machine learning algorithms to predict the character and have used sophisticated hardware setup for recording the accelerometer and gyroscope values. We try to achieve a similar result but with simpler electronics setup and basic machine learning techniques.

SYSTEM DESIGN

The main component of the system is the accelerometer device which one could attach to an actual pen/pencil. This device is connected to an Arduino UNO which reads the accelerometer reading and forward it to the GRT for analysis. The results that are produced by GRT are communicated over to a python server. From where we could have it do different tasks.

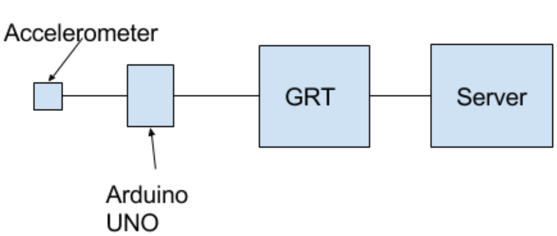


Figure 1

1)Hardware

We used ADLX335 accelerometer which is a 3-axis accelerometer with signal conditioned voltage outputs. The sensor has a full sensing range of +/- 3g.

The accelerometer is connected to the Arduino UNO board where the raw voltage values are collected and using conversion library, these voltage values are converted to sensible values depicting gravitational force which are written to a serial port for further processing.

2) Fabrication

For the ease of moving the device from one pen/pencil to another and to record stable values, we fabricated a box for the accelerometer which can be attached/detached.

We created a design for the box on FreeCAD, and then 3D printed the device. After 3 around iteration we had a proper enclosure for the accelerometer which could be used. Below is the image of the fabricated box.

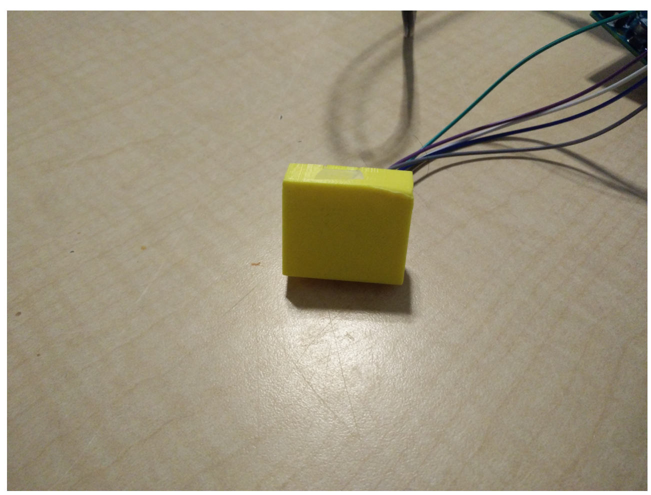


Figure 2

3) Software

Initially we planned on creating an android app which would connect to the device and the results will be shown on the app.

We also explored the idea of using mobile's microphone for gathering data. After struggling to get it right, we pivoted to the idea of desktop application. We developed an initial prototype for or android application as shown below:

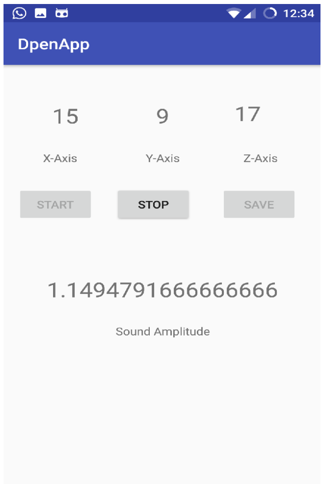


Figure 3

As discussed in system design the reading from the Arduino were used to train a model for recognizing the written character. For the machine learning part, we used GRT toolkit which used DTW for training and detecting the appropriate gestures. GRT has this feature, given by providing a server IP and port number GRT writes the output the corresponding port. So, for reading the GRT results we created a python server which would read the values from the TCP port and display them on the GUI.

Figure 4

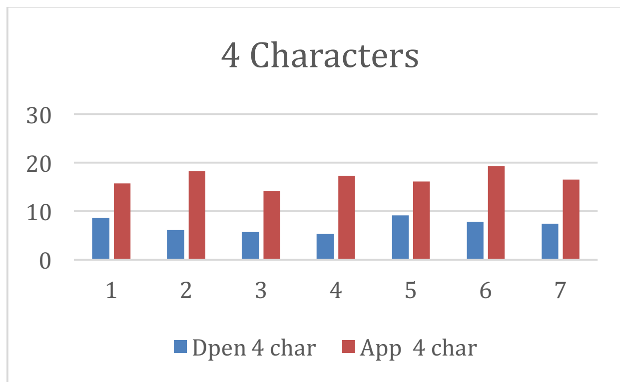
One major problem that we faced while training the machine learning model and collecting samples was that there is very little movement along the 3 axes when we write which make it difficult for the machine learning algorithm to recognize patterns. Which resulted in very little accuracy.

We found out that when we increased the size of written character i.e. writing in the air, the model performed better.

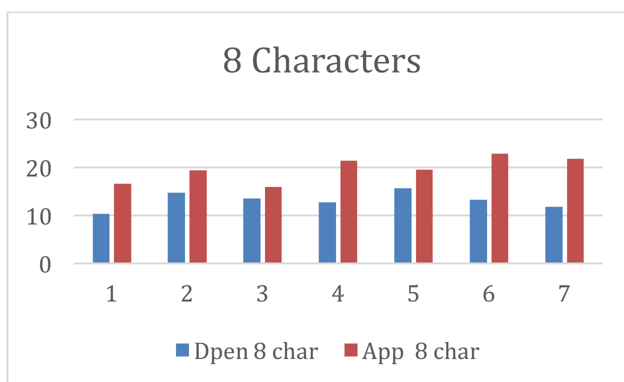
USER STUDY

Since we had trouble developing the prototype to an acceptable level of accuracy, we had to resort to a Wizard of Oz model of testing our prototype. We made users believe that the prototype was indeed working and we generated random errors in the prediction to mock the actual behavior of the prototype. We performed the following test where a user was asked to use our prototype and compare how it fared with using an image processing based android app called Text Fairy to digitize the data by recording the time taken for them to do so in each case. We performed the test in 3 sets varying the number of characters written (4, 8, 12).

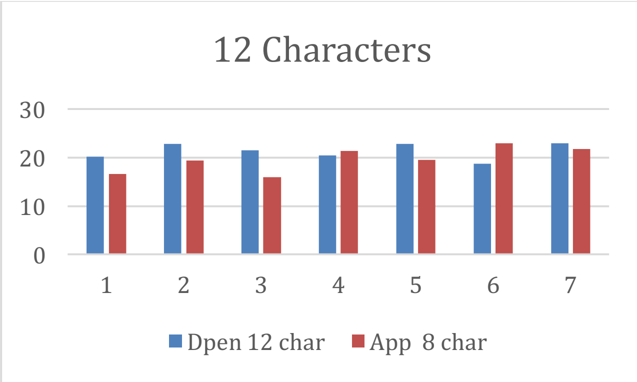
A paired t-test was conducted on each test to check if both the app and our prototype took the same time. The results are as follows:



There was significant difference (p = 0.0000271364) in the time taken by our prototype (M=7.2275, SD=1.2499) and the app (M=16.7083, SD=1.4015) while we tested with 4 characters.



There was significant difference (p = 0.001300989) in the time taken by our prototype (M=13.5958, SD=1.5857) and the app (M=19.0166, SD=2.1905) while we tested with 8 characters.



There was no significant difference (p = 0.001300989) in the time taken by our prototype (M=21.2416, SD=1.5756) and the app (M=21.5333, SD=2.67185) while we tested with 12 characters.

From the results, we suggest that our prototype works better for smaller number of characters as compared to the larger number characters. This is largely due to the amount of errors generated by our prototype during prediction. This could be mitigated using a better algorithm for gesture recognition.

CONCLUSION

Although we had issues with training the data with very little variation, we believe we could use a more advanced machine learning algorithm to effectively train such data and predict the characters with higher success rates. Also, such a tool could also be used for enhancing for other subtle tasks as well.

REFERENCES

1. Arduino tutorials and templates
2. Processing tutorial and templates.
3. J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," IEEE Trans. Ind. Electron., vol. 59, no. 7, pp. 2998–3007,Jul. 2012.
4. Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," IEEE Sensors J., vol. 15, no. 1, pp. 154–163, Jan. 2015.