Air-Recog : Translate your Ideas into Reality

Akshith Nettar Mahalinga - 181IT104
Information Technology
NITK Surathkal
Mangaluru, India
akshithbellare.181it104@nitk.edu.in

Amith Bhat Nekkare - 181IT105
Information Technology
NITK Surathkal
Mangaluru, India
amithbhat01@gmail.com

Kumsetty Nikhil Venkat - 181IT224
Information Technology
NITK Surathkal
Mangaluru, India
nikhilvenkat26@gmail.com

Laharish S - 181IT125
Information Technology
NITK Surathkal
Mangaluru, India
laharish.181it125@nitk.edu.in

***Abstract*—Air-writing refers to virtually writing linguistic characters through hand gestures in three dimensional space with six degrees of freedom. In this project, a generic video camera dependent neural network - based framework has been created. Gestures are performed using a marker of fixed color in front of a video camera. The system identifies the marker and tracks the trajectory of marker tip. A pre-trained neural network is then used to classify the gesture. Two models of the proposed framework have been created, using the popular ML models, Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) respectively, in each instance. A website with a feedback and review section has also been created as an supplement to this framework.**

***Index Terms*—Air-writing, human-computer interaction, gesture recognition, handwritten character recognition, convolutional neural networks.**

I. Introduction

Air-writing systems render a form of gestural Human - Computer Interaction (HCI). Such systems are especially useful for building advanced user-interfaces that do not require tradition mechanisms of linguistic input such as pen-up - pen-down motion, hardware input devices or virtual keyboards by providing an interface for writing through hand gestures in three dimensional space with six degrees of freedom. Unlike conventional writing air-writing systems lack actual anchoring and reference position on writing plane, gestures are guided by considering imaginary axes in three dimensional space. Essentially these facts contribute to the increased variability of writing patterns in such scheme and thereby account for the non-trivial nature of the problem.

With rapid development of depth sensors in recent years such as Kinect and LEAP Motion, the possibility of air-writing systems has been emerged. Depth sensors along with computer vision techniques are used to track finger tips followed by recognition of the performed gestures using a trained model. But these sensors are not widely available to common devices which restricts these systems to be easily accessible. Hence, programs and technologies using commonly available hardware such as cameras are imperative for air-writing systems to be a mass phenomenon.

Furthermore, such air-writing systems can be useful in various practical scenarios. Some typical use cases include:

• A doctor is taking notes regarding a patient while performing an operation and therefore unable to use conventional methods of writing.

• Composing message on a mobile device in a more interactive way even if the key or touch input of such device is malfunctioning.

• Teaching alphabets and digits to children and adult learners in an interactive and fun manner.

• Helping patients in rehabilitation from hand injuries to exercise and recover motor skills.

II. Literature Survey

We have taken [I] to be our base paper for this mini project. It proposes a method to use a generic video camera to capture motion of a marker in three dimensions and ”read” it to identify it as a letter or digit of English language.

Previous works to address air-writing problem generally relied upon depth sensors such as Kinect [1] and LEAP Motion [2], or wearable gesture control and motion control device such as Myo [3], or multi-camera setup to estimate depth information. While these approaches account for easier tracking and better accuracy, they suffer from cost-effective general purpose usage due to essential dependency on the external hardware.

Chen et al. [4], [5] used LEAP Motion for tracking and a Hidden Markov Model (HMM) for recognition. They

reported 0.8% error rate for word-based recognition and 1.9% error rate for letter-based recognition. Kristensson et al. [6] proposed a bimanual markerless interface for depth sensors using a probabilistic scale and translation invariant algorithm. They achieved 92.7% and 96.2% accuracy for one-handed and twohanded gestures respectively. Dash et al. [7] employed Myo armband sensor along with a novel Fusion model architecture by combining one Convolutional Neural Network (CNN) and two Gated Recurrent Units (GRU). The Fusion model is reported to outperform other widely used models such as HMM, SVM, KNN and achieved an accuracy of 91.7% in a person independent evaluation and 96.7% in a person dependent evaluation.

The drawback of the above mentioned works is that all such previously proposed systems involves either special-purpose sensors or a multi-camera setup which restricts mainstream adoption of these systems. In this project a single generic video camera based air-writing system is proposed which can be implemented on any commonly used devices (such as smartphone, laptop etc.) with a in-built video camera.

III. Problem Statement

To implement a web-based single generic video camera based air-writing system, compatible with any commonly used device (such as smartphone, laptop etc.) with a in-built video camera. This is to be supplemented by a web application and a review system, both of which follow HCI guidelines.

IV. Methodology

In this section, we describe the methods used in this project.

A. *Multi-layer Perceptron Model*

We use Python's MNIST[9] library to load the data. We get the data ready to be fed to the model. Splitting the data into train and test sets, standardizing the images and other pre-processing the dataset. In Keras, models are defined as sequence of layers. We first initialize a sequential Model and then we add the layers with respective neurons, activation functions, and dropout functions in between the layers in the model.

The model, as expected, takes 28 x 28 pixels this is done by flattening out the image and passing each of the pixel in a 1-D vector as an input. The output of model has to be a decision on one of the letters, so we set the output layer with 26 neurons where the decision is made in probabilities.

Now that the model is defined, we can compile it. Compiling the model uses the efficient numerical libraries under the covers the back-end which is built in TensorFlow.

Here, we specified some properties that were needed to train the network. By training, we are trying to find the best set of weights to make the decision on the input. We must specify the loss function to use to evaluate a set of weights, the optimizer used to search through different weights for the network and any optional metrics we would like to collect and report during training.

Here, we train the model using a model check pointer, which will help us save the best model in terms of metrics.

B. *Convolutional Neural Network Model*

These two steps are exactly the same as the steps we implemented in building the Multi-layer Perceptron Model.

Convolutional Neural Networks are very similar to ordinary Neural Networks they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity.

ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

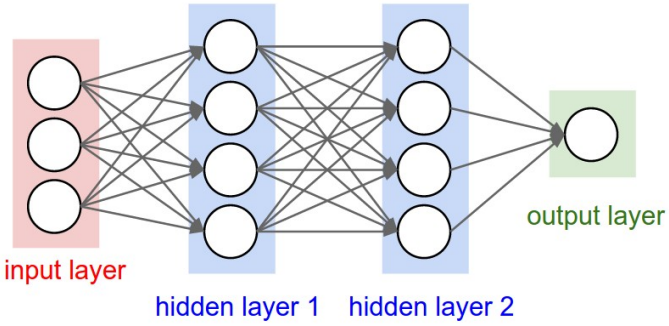

Fig. 1. ConvNet Architecture

dropout is introduced as regularization in our model to reduce over-fitting problem. While compiling the model we used categorical-crossentropy loss function as it is a multi-class classification problem. Since all the labels carry similar weight we prefer accuracy as performance metric. A popular gradient descent technique called AdaDelta is used for optimization of the model parameters.

The model fitting is same as Multi-layer Perceptron Model.

C. *Initializing the model*

we load the models built in the previous steps. We then create a letters dictionary, blue-Lower and blue-Upper boundaries to detect the blue bottle cap, a kernel to smooth things along the way, an empty blackboard to store the writings in white like the alphabet in the EMNIST dataset, a double ended queue was used to store all the points generated by the pen or a blue bottle cap, and a couple of default value variables.

D. *Capturing the air writings*

Once we start reading the input video frame by frame, we try to find the blue bottle cap and use it as a pen. We use the OpenCV's 'video capture' method method to read the video, frame by frame using a while loop, either from a video file or from a webcam in real time. In this case, we pass 0 to the method to read from a webcam. The following code demonstrates the same.

Once we start reading the webcam feed, we constantly look for a blue color object in the frames with the help of 'range' method and use the blue-Upper and blue-Lower variables initialized beforehand. Once we find the contour, we do a series of image operations and make it smooth. Smoothing just makes our lives easier. If you want to know more about these operations — erode, morph and dilate, check this out.

Once we find the contour i.e. the if condition passes when a contour is found, we use the center of the contour in this case a blue pen or object to draw on the screen as it moves.

Then we check if a contour is found and if yes, it takes the largest one , draws a circle around it using the 'minimum enclosing circle' method, get the center of the contour found with the help of 'moments' method. In the end, the center is stored in a double ended queue called 'points' so that we can join them all to form a complete writing.

We display the drawing on both frame and blackboard. One for external display and the other to pass it to the model.

E. *Scraping the air written character and passing it to the model*

Once the user finishes writing, we take the points we stored earlier, join them up, put them on a black board and pass it to the respective models.

The control enters this else-if block when we stop writing, since there were no contours detected. Once we verify that the points double ended queue is not empty, we are now sure that the writing is done. Now we take the blackboard image, do a quick contour search again to scrape the writing out. Once detected and confirmed, we cut it appropriately, resize it meet the input dimension requirements of the models we built that is 28 x 28 pixels and pass it to the respective models.

F. *Showing model predictions*

We then show the predictions made by our models on the frame window. And then we display it using the 'image show' method. After falling out of the while loop we entered to read data from the webcam, we release the camera and destroy all the windows.

V. Results and Analysis

In this section, we will discuss the different components of the web application created for this project.There are three main components of this website : the landing page, the air writing demo, and the review page.

A. *The Landing Page*

The landing page is the first page to be visible when a person visits our web application (it can be found at *https://air-recog.web.app/*). The MERN stack technology was used to create the landing page.

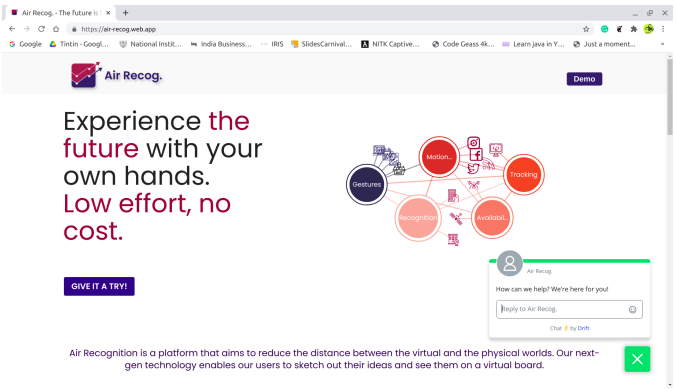

Fig. 2. Home Page : View 1

In the website, we have added several elements related to human - computer interaction. For example :

• Inclusion of a chat bot to provide users with real-time responses to any queries.

• Attractive and aesthetic design of website to entice potential users to use the product. For example, we have an interactive info graph with the main concepts and ideas of our project adorning the main page. The nodes can be moved around with the mouse as well to give the result of natural-like movement, which has a pleasing effect and enhances the look of the web application.

• A bigger size button redirecting to the air-writing demonstration. This is in accordance with Fitts Law which says that as the button size increases, the time required to click it decreases. In addition, the button has been added near the center of the screen enabling quicker access.

• As the user scrolls down, they can also see the several benefits and advantages of using the air writing system. This would motivate them to try and use the product as well, in various situations.

• Finally, they can see several key statistics, such as the high accuracy rate, extremely low reaction time and the zero cost associated with our product. Finally, the footer bar with the contact links and social media profiles gives a polished look and feel to the website.

B. *Demonstration Page*

The 'Give it a try' button in the landing page redirects the user to the demo page, where they can try out the product we

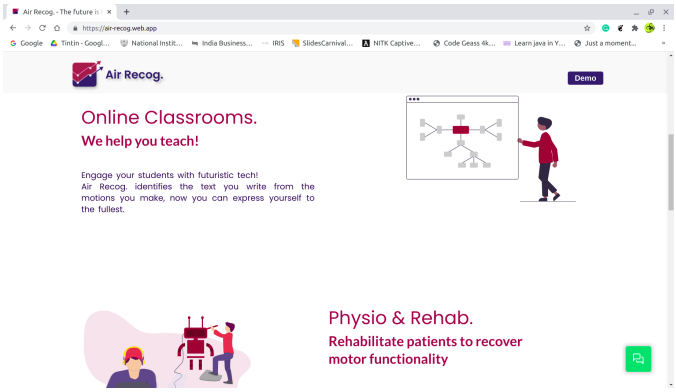

Fig. 3. Home Page : View 2

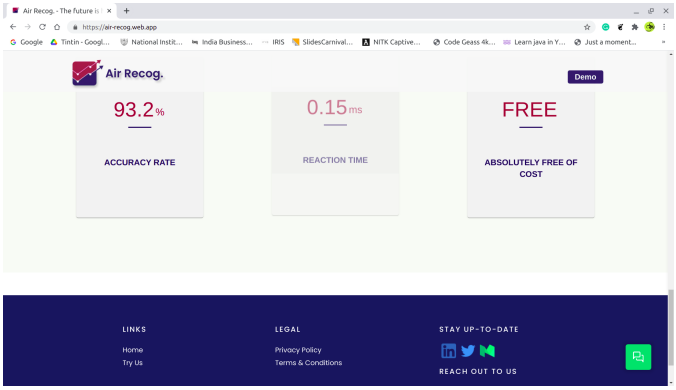

Fig. 4. Home Page : View 3

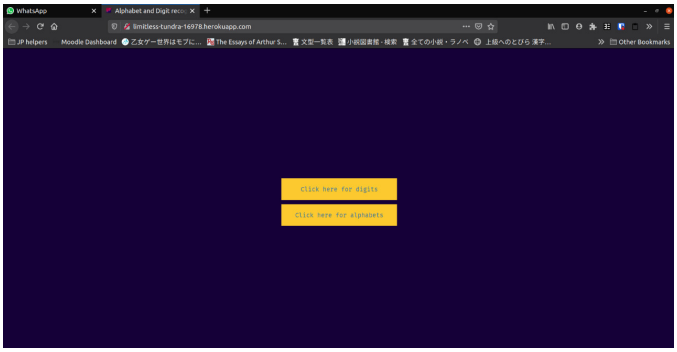

Fig. 5. Demo Page

have created. This has been deployed to the web application as a Flask application.

We have given two options to the user : they can click on the digit option to initialize the pre - trained model to detect digits. Similarly, the alphabet option can be clicked for the alphabet detection model. Below are the screenshots of the respective models :

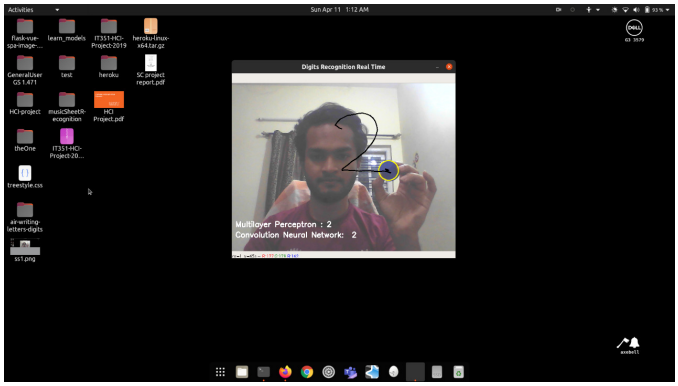

Fig. 6. Digit Recognition

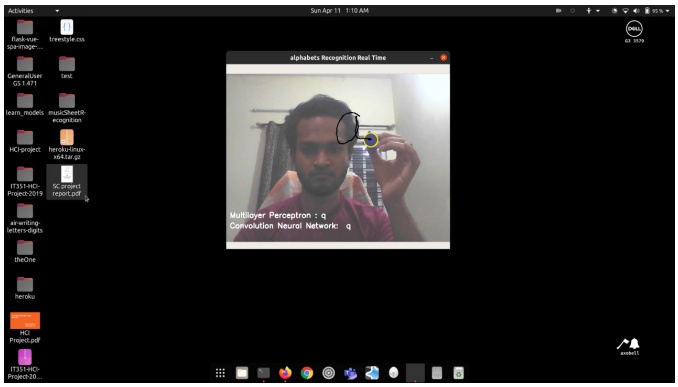

Fig. 7. Alphabet Recognition

C. Review Page

After the users have tried out the air writing system, we have given them an option to give a review of the experience. It is important to note that the review is given completely voluntarily(by giving them option of clicking a 'go to review' button, and is not forced upon the users. We have done this to enhance user experience with the product.

For the actual survey, we have leveraged the use of a free, high quality survey tool called Survey Monkey. We set 8 questions which would allow us to analyze the user's experience with the air writing product. Below are the screenshots from the review page:

VI. Results and Analysis

Upon analysis of the models, we found that it is effective around 93.2% of the time on average. This is quite high since we utilize only a single camera for motion capture.

We circulated this website among 47 people from our friends circle and close family members. We wanted to check if there existed any relationship between the age of the user and the rating which they gave, and if there existed any relationship between the gender of the user and the rating which they gave. For this purpose, we performed the

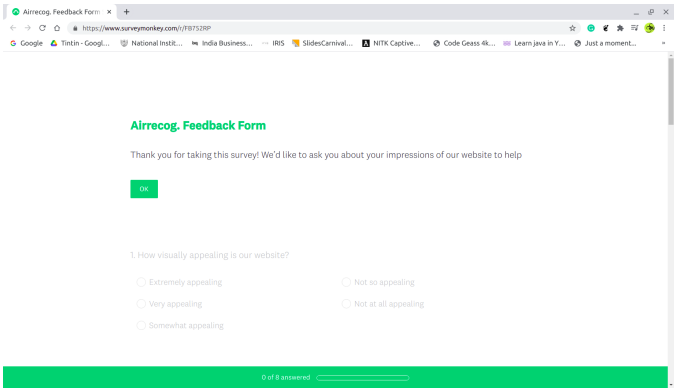

Fig. 8. Review Page : View 1

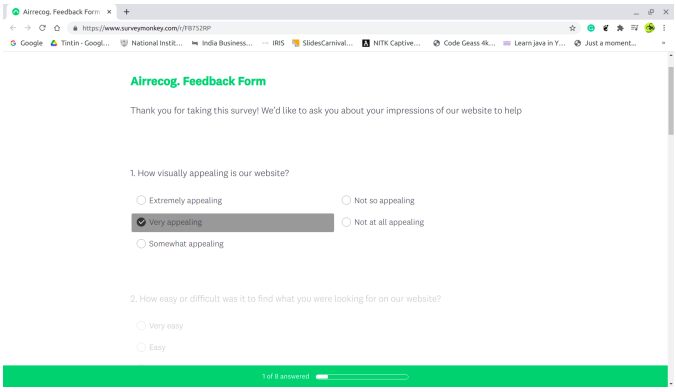

Fig. 9. Review Page : View 2

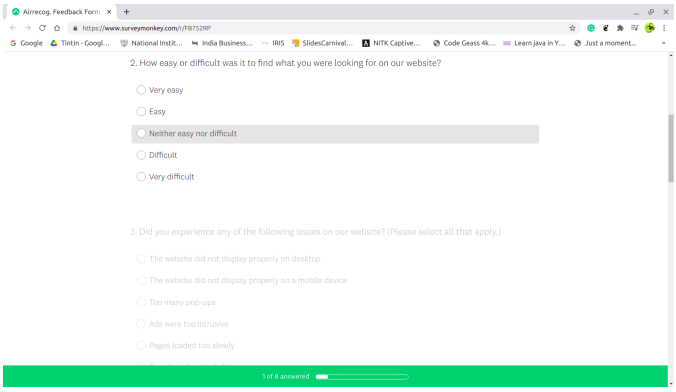

Fig. 10. Review Page : View 3

Chi-Square Test of Independence.

The Chi-Square Test of Independence (a type of F-Test) is used to check if two categorical variables are independent of each other or not. Here, the two categorical variables were the age groups(or the gender), and the sentiment of the rating (positive, neural, negative) respectively. Below are the results of the respective Chi-Square Tests.

For the age group vs ratings test, we made the Null Hypothesis that the age groups and ratings variables are

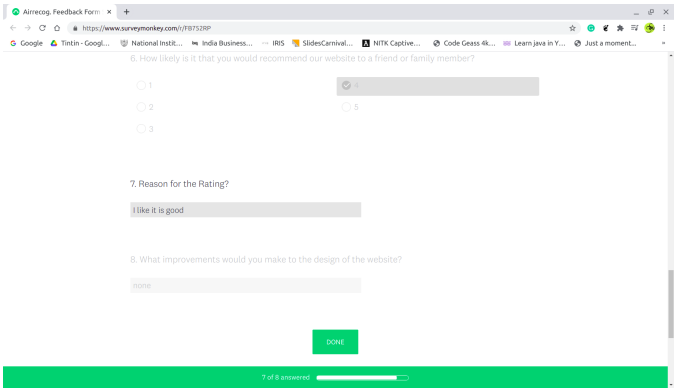

Fig. 11. Review Page : View 4

Results						
	negative	neutral	positive			***Row Totals***
<18	2 (1.91) [0.00]	1 (1.40) [0.12]	3 (2.68) [0.04]			6
18<=x<=30	2 (5.43) [2.16]	5 (3.98) [0.26]	10 (7.60) [0.76]			17
30<=x<=50	1 (2.87) [1.22]	3 (2.11) [0.38]	5 (4.02) [0.24]			9
x>=50	10 (4.79) [5.68]	2 (3.51) [0.65]	3 (6.70) [2.04]			15
Column Totals	15	11	21			**47 (Grand Total)**

The chi-square statistic is 13.553. The *p*-value is .035048. The result is significant at $p < .05$.

Fig. 12. Results of Age vs Rating Chi-Square Test

independent of each other. Similarly, the Alternate Hypothesis was that the two variables are dependent on each other.

From the table, it is clear that the Chi-Square statistic is 13.553. Also, the p-value is 0.035, which is less than the significance value of 0.05. Hence, the Null Hypothesis is rejected and we prove that the age group that a user belongs to has an effect on the ratings they give.

Results						
	Negative	Neutral	Positive			***Row Totals***
Male	5 (4.94) [0.00]	9 (8.64) [0.02]	15 (15.43) [0.01]			29
Female	3 (3.06) [0.00]	5 (5.36) [0.02]	10 (9.57) [0.02]			18
Column Totals	8	14	25			**47 (Grand Total)**

The chi-square statistic is 0.0724. The *p*-value is .96447. The result is *not* significant at $p < .05$.

Fig. 13. Results of Gender vs Rating Chi-Square Test

For the gender vs ratings test, we made the Null Hypothesis that the gender and ratings variables are independent of each other. Similarly, the Alternate Hypothesis was that the two variables are dependent on each other.

From the table, it is clear that the Chi-Square statistic is 0.0724. Also, the p-value is 0.964, which is greater than the significance value of 0.05. Hence, the Null Hypothesis is not rejected.

Upon analysis of the reviews given by 47 users of our product, we found that nearly 69% had a positive impression of our product (they gave 4 or 5 star rating), 23% had a

neutral experience (3 star rating) and only 8% had a negative experience (1 or 2 star rating).

Further, upon performing exploratory data analysis of the comments received, we found that nearly 78% of them contained the phrase ”good product”, ”nice”, ”great” and other positive comments. Hence, we can without a doubt say that users have enjoyed using our product.

Next, we perform the user activity analysis. The data for the below graphs was obtained from the web application’s Firebase console and Google Analytics.

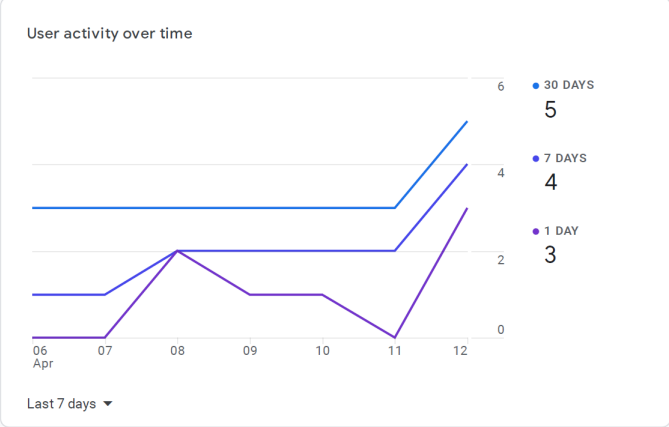

Fig. 14. Analysis of active users

Fig. 14 shows what is the trend of users in our application with respect to each day.

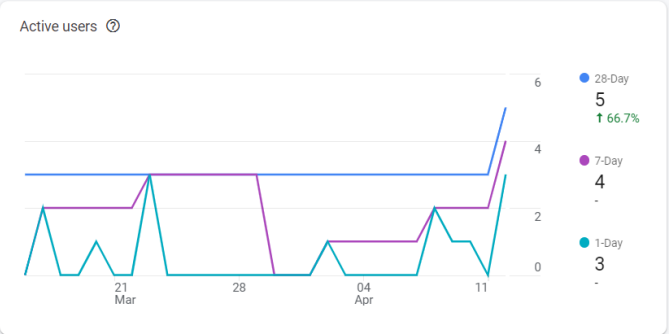

Fig. 15. User analysis

Fig. 15 shows the analysis of users in our application.The novice and expert users can be identified from this graph. It is also evident that an expert users activity is more streamline which means those users are more interactive with our application.

Fig. 16 helps us understand how well our application retains it’s users over the past few weeks. It can be observed that almost all of the users are interacting with our application

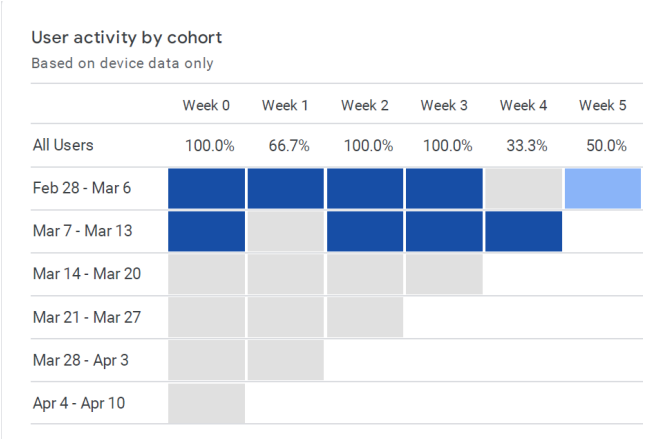

Fig. 16. Analysis of the retention of users in our application

and are frequently using it.

Finally, we perform the task analysis on the website by measuring the number of scrolls and clicks required by the user to get to the demo of the air writing model.

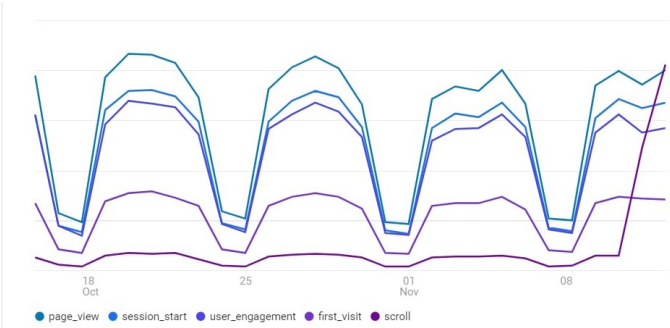

Fig. 17. Task analysis on our application

We observe that the users on average perform very less number of tasks such as scrolls, clicks , page views to use the product.

VII. Conclusion

In this project, we have implemented a web application to demonstrate our generic video camera based air-writing system. It is compatible with any device with a in-built video camera, such as smartphone, laptop, etc.

The web application also includes a landing page with details about air writing systems and their advantages, and a review page to collect, understand and analyse user feedback. All three components of the web app follow HCI guidelines and principles such as Fitts Law, etc.

Upon performing the Chi-Square Test of Independence, we found that there was a negative relationship between the

age group of the user and the rating which they gave for the application.

Upon analysis of the information we collected from user feedback, we found that nearly 69% of users liked our product. Also, 78% of people gave positive comments in their reviews. These results are corroborated by the user and task analysis which we have done, which indicate quick access to the product and increasing number of views which our web application has. Hence, we have built a successful air-writing system using HCI principles and guidelines.

VIII. Acknowledgements

We would like to thank our teacher for the course, Prof. Ram Mohana Reddy Guddeti for his invaluable feedback and advice during the genesis of this project. We would also like to acknowledge Ms. Trupti Chandak for her constant support, encouragement and help.

IX. Individual Contribution

While all the members were significantly involved with the project, and we frequently collaborated with one another for debugging and problem solving, following is the distribution of work we did :-

Amith Bhat : Creation of the Digit CNN model for the air writing system and implemented mathematical models on the website,and Empirical Analysis of the website. - 25%

Laharish S : Creation of the Digit MLP model for the air writing system, built flask components, and deployed the application. - 25%

Akshith Bellare : Creation of the Alphabet MLP model for the air writing, built flask components, and deployed the application. - 25%

Nikhil Venkat : Creation of the Alphabet CNN model for the air writing system, created landing page using Human-Computer Interaction design principles and Guidelines, and implemented mathematical models on the website. - 25%

The Gantt chart is given below :

Fig. 18. Gantt Chart

X. Base Paper

[I] P. Roy, S. Ghosh and U. Pal, "A CNN Based Framework for Uni-stroke Numeral Recognition in Air-Writing," 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 2018, pp. 404-409, doi: 10.1109/ICFHR-2018.2018.00077.

References

[1] Microsoft Corporation. Kinect. 2010. URL: https://developer.microsoft.com/en-us/windows/kinect/.

[2] Leap Motion Inc. LEAP Motion. 2010. URL: https://www.leapmotion.com/.

[3] Thalmic Labs Inc. Myo. 2013. URL: https://www.myo. com/.

[4] M. Chen, G. AlRegib, and B. H. Juang. "Air-Writing Recognition–Part I: Modeling and Recognition of Characters, Words, and Connecting Motions". In: IEEE Transactions on Human-Machine Systems 46.3 (June 2016), pp. 403–413. ISSN: 2168-2291. DOI: 10.1109/ THMS.2015.2492598.

[5] M. Chen, G. AlRegib, and B. H. Juang. "Air-Writing Recognition–Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data". In: IEEE Transactions on Human-Machine Systems 46.3 (June 2016), pp. 436–444. ISSN: 2168-2291. DOI: 10. 1109/THMS.2015.2492599.

[6] P. O. Kristensson, T. Nicholson, and A. Quigley. "Continuous recognition of one-handed and two-handed gestures using 3D full-body motion tracking sensors". In: Proceedings of the 2012 ACM international conference on Intelligent User Interfaces. ACM. 2012, pp. 89–92.

[7] A. Dash, A. Sahu, R. Shringi, et al. "AirScript–Creating Documents in Air". In: 14th International Conference on Document Analysis and Recognition. 2017, pp. 908– 913.

[8] A. Schick, D. Morlock, C. Amma, et al. "Visionbased Handwriting Recognition for Unrestricted Text Input in Mid-air". In: Proceedings of the 14th ACM International Conference on Multimodal Interaction. ICMI '12. ACM, 2012, pp. 217–220.

[9] Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST Database of Handwritten Digits. 1998.