IMPROVED HANDS-FREE TEXT ENTRY SYSTEM

ADVISOR: DR. CHRIS POLLETT

COMMITTEE MEMBERS:

1. DR. KATERINA POTIKA

2 MR. KEVIN SMITH.

BY: GAURAV GUPTA

AGENDA

1. Introduction
2. Background
3. Design and Architecture
4. Implementation
5. Experiments
6. Conclusion and Future Work
7. References

INTRODUCTION

Modes of inputs can be distinguished based on medium like:

1. Voice based inputs (For ex: Siri, Alexa, Google Assistant)
2. Video based inputs
3. Touch based inputs

Improved Hands-Free Text Entry System (IHFTES) provides a novel technique to generate text characters by capturing rigid and non-rigid motions of face.

Proposed solution will help people with certain disabilities to interact with computer systems.

Placement of characters in IHFTES is based on the frequency of characters.

BACKGROUND

1. Techniques suggested by Adam Nowosielski in [1] and [2].
   a. Three-Step keyboard[1].
   b. Two-Step Keyboard
2. Face recognition strategies
   a. Face detection based on skin color and Adaboost algorithm
   b. Face detection based on Adaboost and new Haar-Like features
   c. Face detection using neural networks
3. Tensorflow
4. Atom Editor

Implementation By Adam Nowosielski

Adam Nowosielski proposed two novel methods to generate text using head movements

a. Three-step keyboards[1]
b. Two-letter key keyboards[2]

Adam Nowosielski captures rigid motions of head (motions in vertical and horizontal direction).

Nowosielski constrained the motion in four direction (Up, Down, Right and Down)

Three-Step Keyboard

Nowosielski suggest since, motion is constrained to four direction, the number of choices are limited to four.

Four choices are not enough to cover all possibilities including alphabets-symbols, digits, and other punctuation-symbols.

Nowosielski suggested three head movements to cover all possibilities.

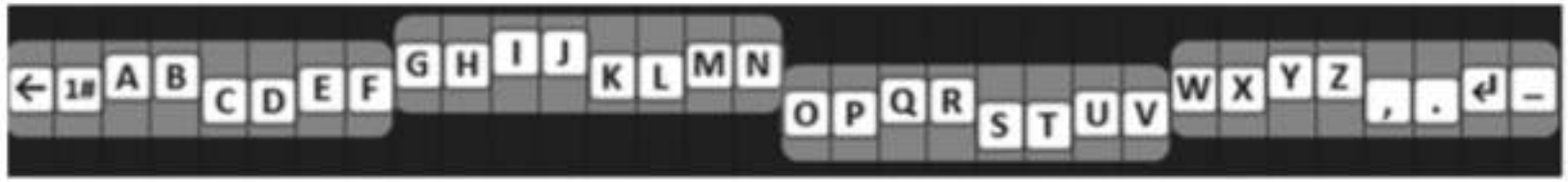


Fig. 1. Two Step Keyboard Layout[1][2]

Characters are arranged in four groups, two on same level (one on the extreme left and another on the extreme right), shown in Fig. 1.

Other two groups are arranged such that one is in up direction and another in down direction.

Three-steps keyboard character selection example

Fig. 2. Example of selecting character ‘A’ using Nowosielski [1][2]

Two-Letter Key Keyboard

Nowosielski suggests using two letter key keyboard, character can be reached in two head movements instead on three head movements.

Two-letter key keyboard supports dictionary feature which enables user to select character from the dictionary word.

Two letter key keyboard provide back compatibility to three letter key keyboard.

Fig. 3. Layout of Two-letter Key Keyboard [2]

Two-Letter Key Keyboard in action

Fig. 5. shows the usage of two-letter key keyboard. The word in suggestion are generated using an English dictionary compiled.

The words are formed using the movements LD -> DL -> UR -> DL -> DR

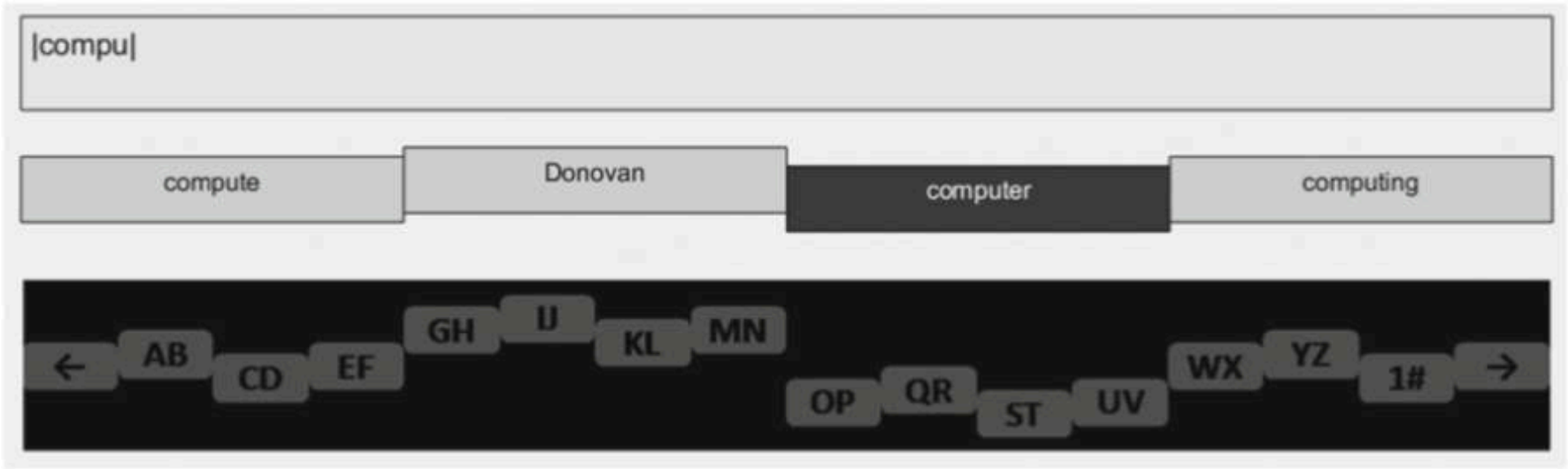


Fig. 5. Two Letter Key Keyboard in action

Face Recognition Strategies

Face recognition strategies help in identifying the facial region in an image.

To detect head movements it is vital to identify the facial region of the person of interest.

Following are the face recognition strategies studied:

1. Face detection based on skin color and Adaboost algorithm[3]
2. Face recognition based on Adaboost and new Haar-Like feature[4]
3. Face detection using neural network[5]

Face Detection Using Neural Networks[5]

Author using a convolutional neural network to identify a facial region in an image.

Author divides the facial recognition in two stages:

1. Several neural network based filter are tried to determine the presence of face
2. Merge overlapping frames together to detect a face in the region.

Author checks a 20x20 window to determine whether a face is present in the current window or not.

Author reduces the size of image several times at different scales in order to identify the presence of face which might be smaller than 20x20 image window.

Author pass this window to a neural network to train and identify the presence of facial region.

Face Detection Using Neural Networks[5] (contd.)

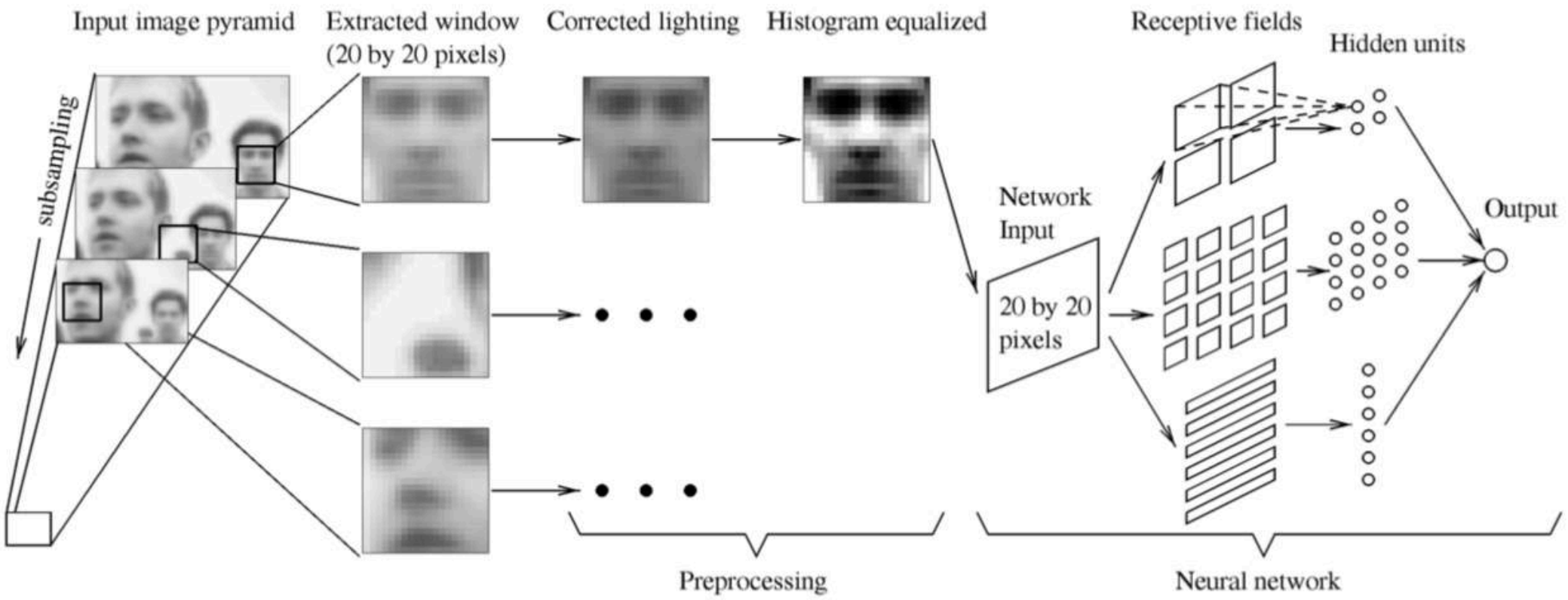


Fig. 6. Basic algorithm to detect face in neural networks[5]

Tensorflow [6]

Open-source platform developed by Google Brain Team.

Provides flexibility to develop complex machine learning algorithm with ease.

Provides flexibility to run machine learning tasks over Nvidia's Graphical Processing Unit (GPU)

Tensorflow Object Detection API [7]

Provides pre-trained models and configurations to detect several objects in advance.

Provides flexibility to train for customizable objects.

Provide the flexibility to users to write customizable configuration file.

Atom Editor[9]

Atom is a hackable and configurable cross-platform IDE.

User can use prebuild extension or develop one from scratch.

Atom editor is an open-source platform.

It provides the flexibility to add or choose from more than 10,000 pre-built extension

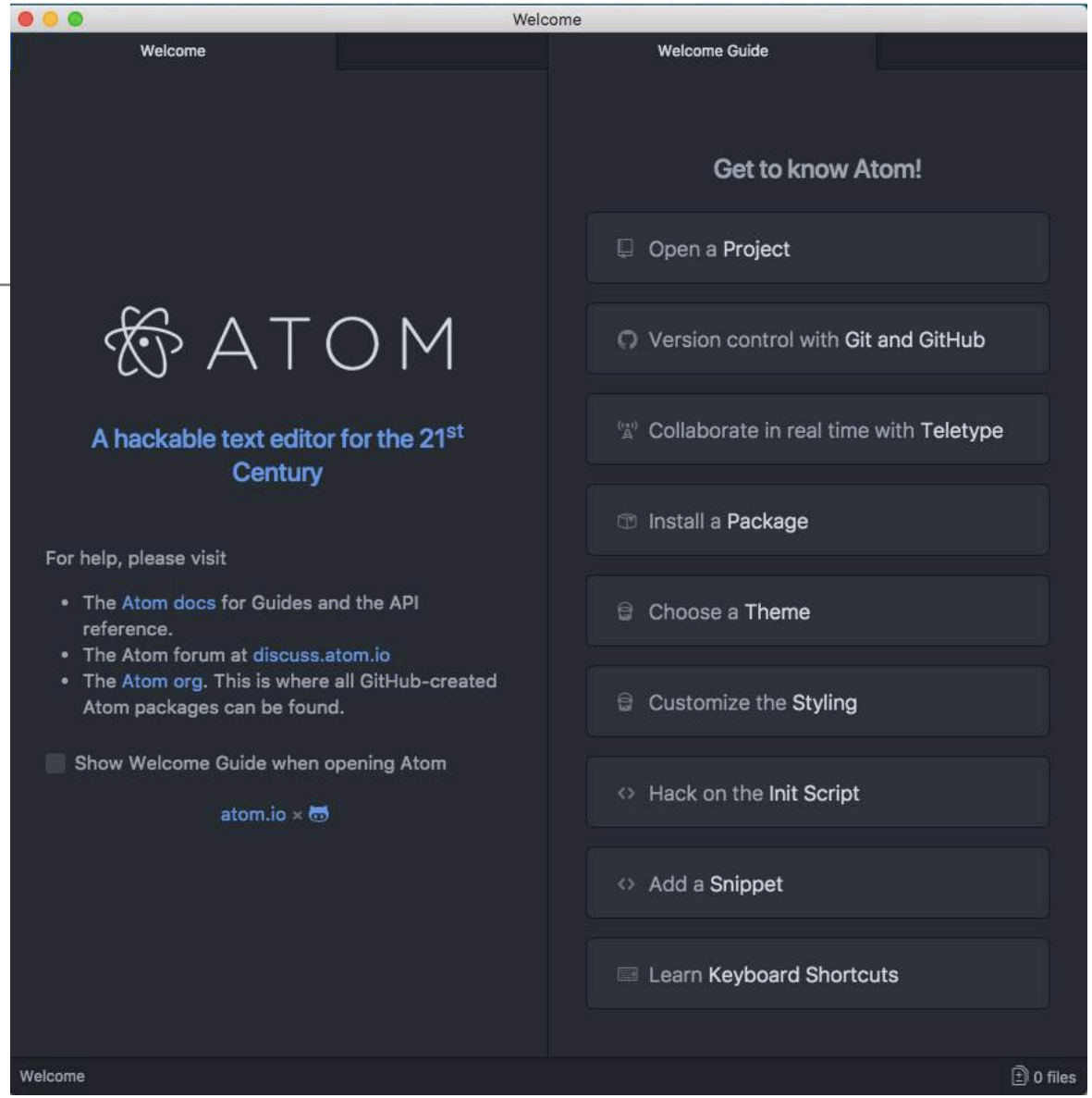


Fig. 7. Layout of Atom on launch of application

DESIGN AND ARCHITECTURE

Design of IHFTES is modal.

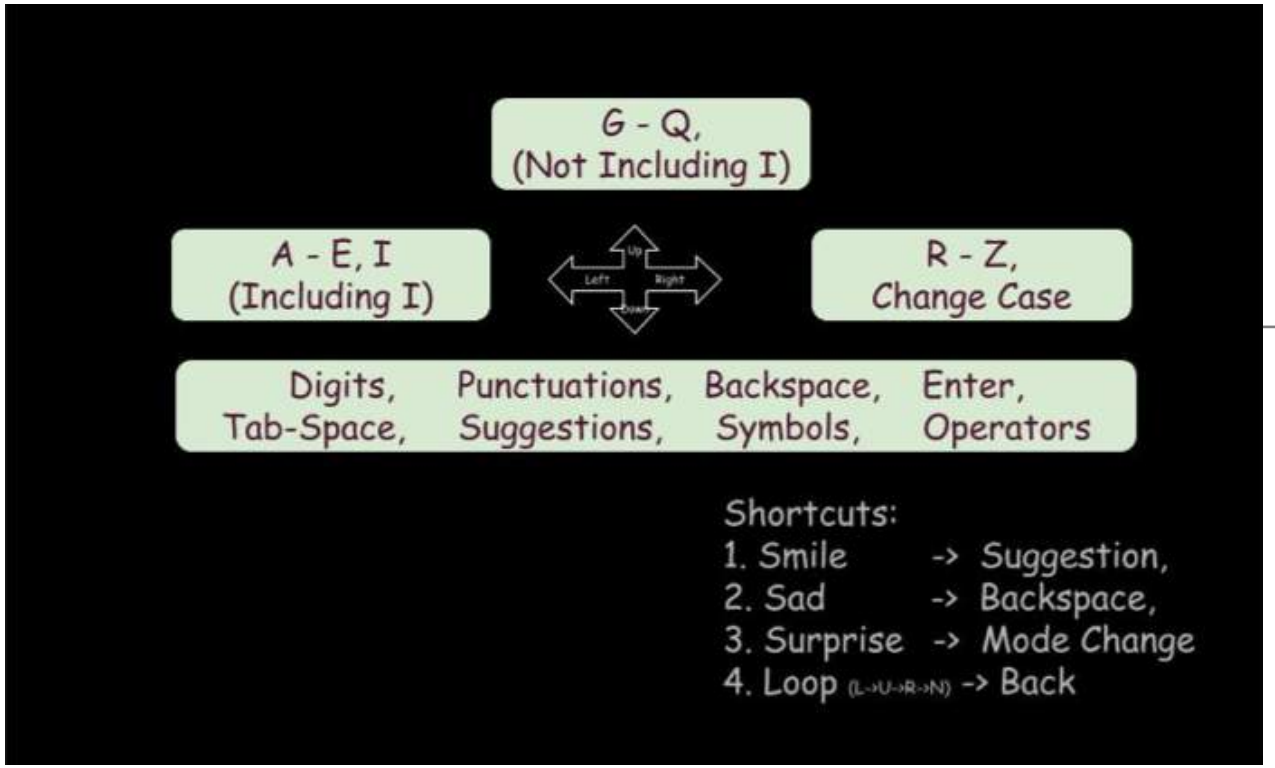
There are two modes specifically:

1. Alphabetical mode
2. Numerical mode

Placement of alphabet-symbols is based on frequency of alphabet-symbols in English dictionary.

IHFTES captures rigid and non-rigid motions.

Fig. 8. shows the layout of alphabetical mode and numerical mode.



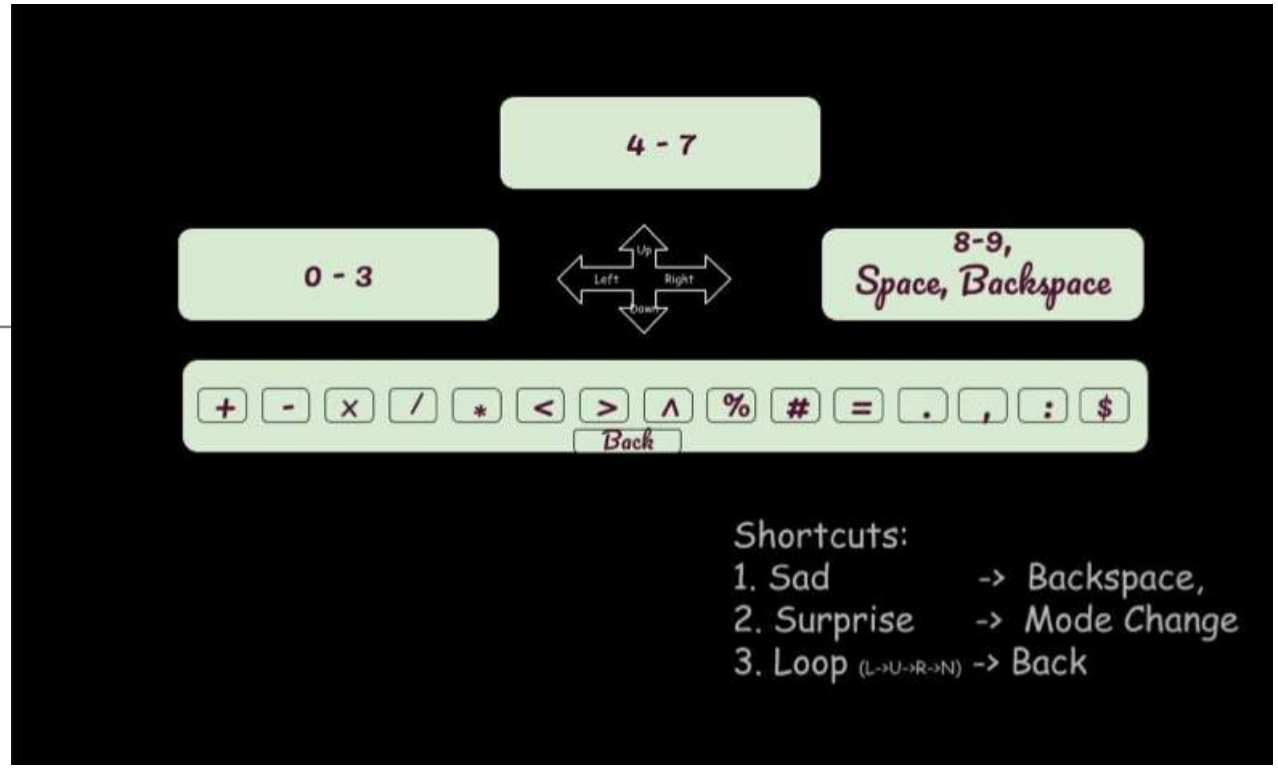


Fig. 8. Layout of alphabetical mode and numerical mode.

Second level layout details of IHFTES

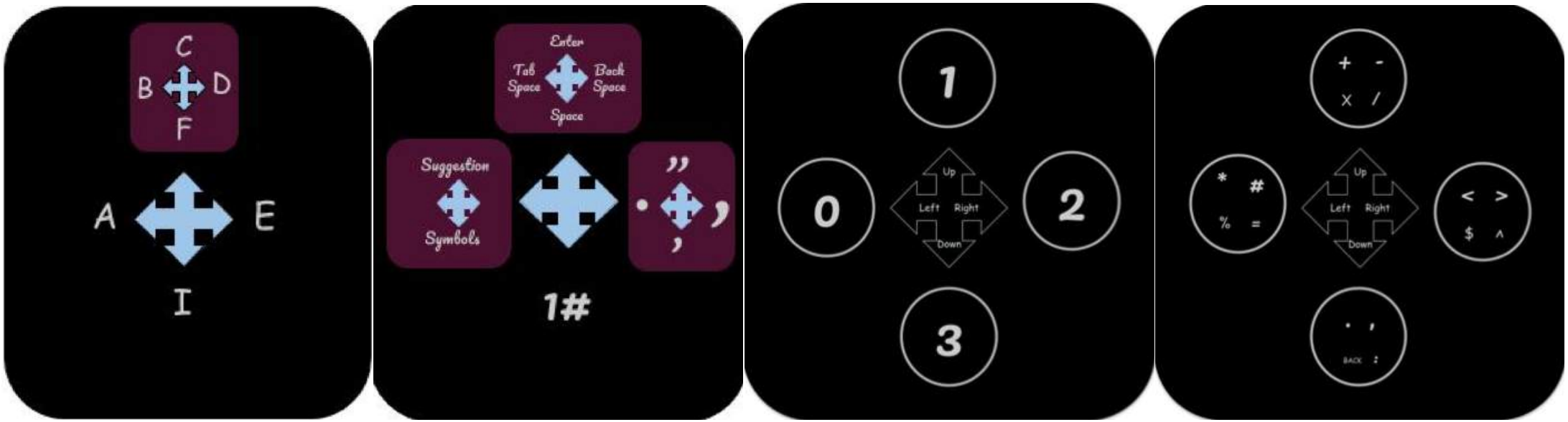


Fig.9. Second level layout details of IHFTES

Third level layout details of IHFTES

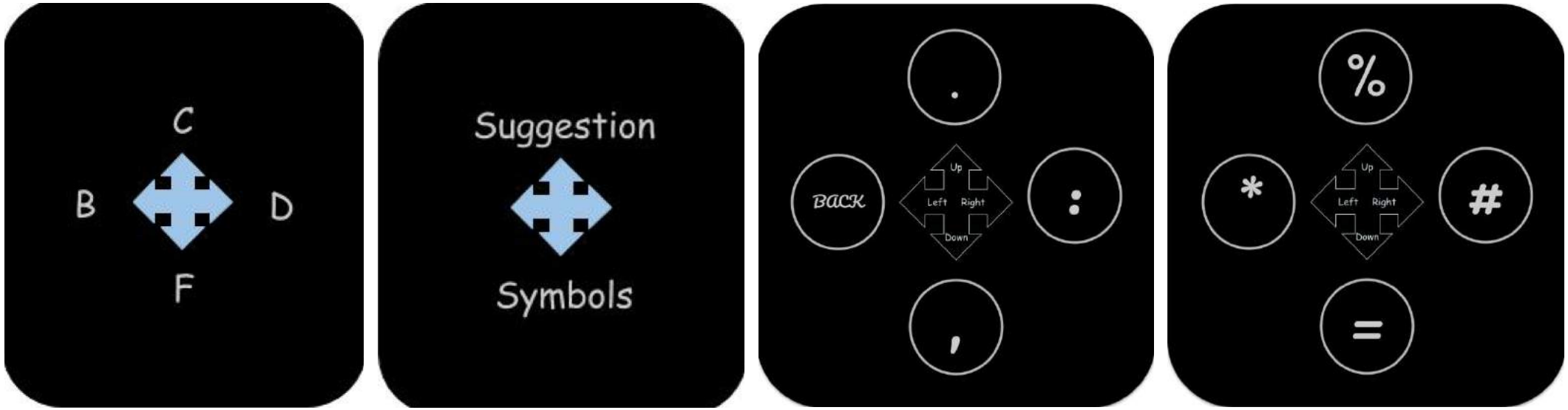


Fig. 10. Third level layout of IHFTES

IMPLEMENTATION

Implementation of IHFTES is dependent on several sub-components.

Different subcomponents built for IHFTES are:

1. Key-logger tool
2. Auto-Reload extension for Atom editor
3. Word guessing tool
4. Facial gesture detection
5. Head movement detection

Key-Logger Tool

Key-logger tool developed in Python 2.7 programming language.

Tool was developed with the focus on monitoring what key has been pressed.

A key pressed and released is shown on terminal. The entire content is shown in atom editor.

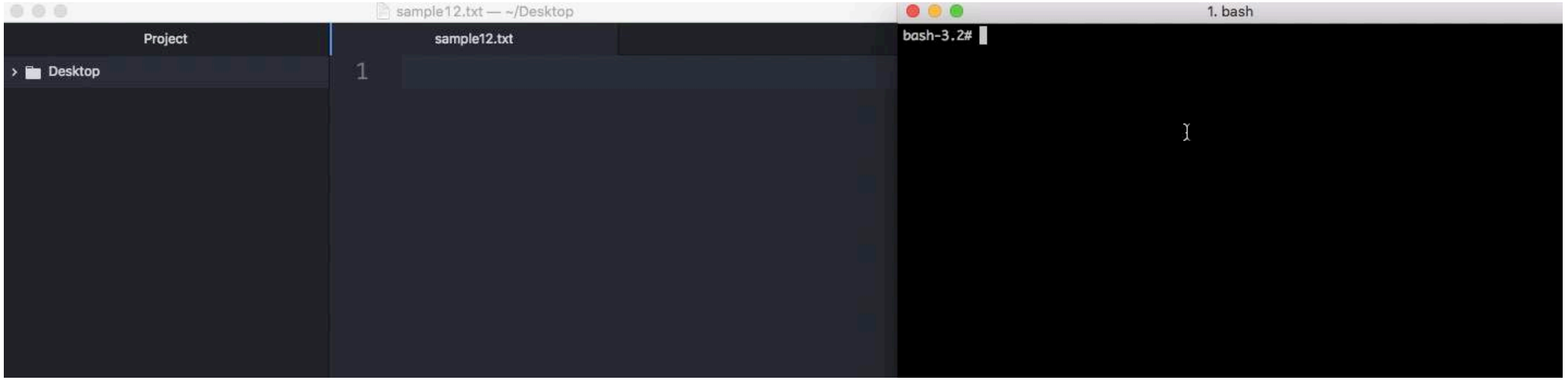


Fig. 11. Output of Key-logger tool

Auto-Reload: extension for Atom Editor

Auto-Reload extension is a tool developed for atom editor.

Tool continuously monitors all the files that are opened in the Atom editor.

If the content of the file is changed from some other medium it will be reflected back on the editor.

Extension can be enabled from the packages sub-menu or using the keyboard shortcut.

Fig. 12. Auto-Reload extension for Atom editor

Word Guessing Tool

Word guessing tool guess the word based on partial word typed.

If no word is provided as input, the suggestion would be based on shortest word available in dictionary.

Number of choices for words is constrained to four.

English dictionary has been fetched form [10], dictionary is further filtered to hold valid words.

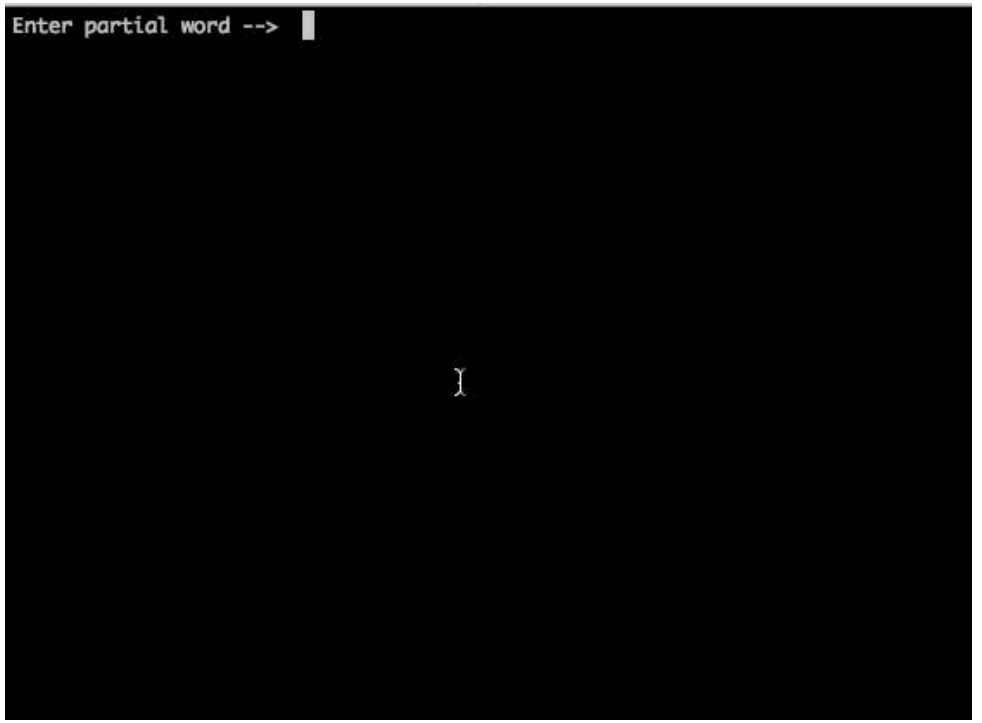


Fig. 13. Word Guessing Tool

Facial Gesture Detection

Facial gesture detection tool is implemented to capture non-rigid motions made by face.

Convolutional neural network is used to detect the face gestures.

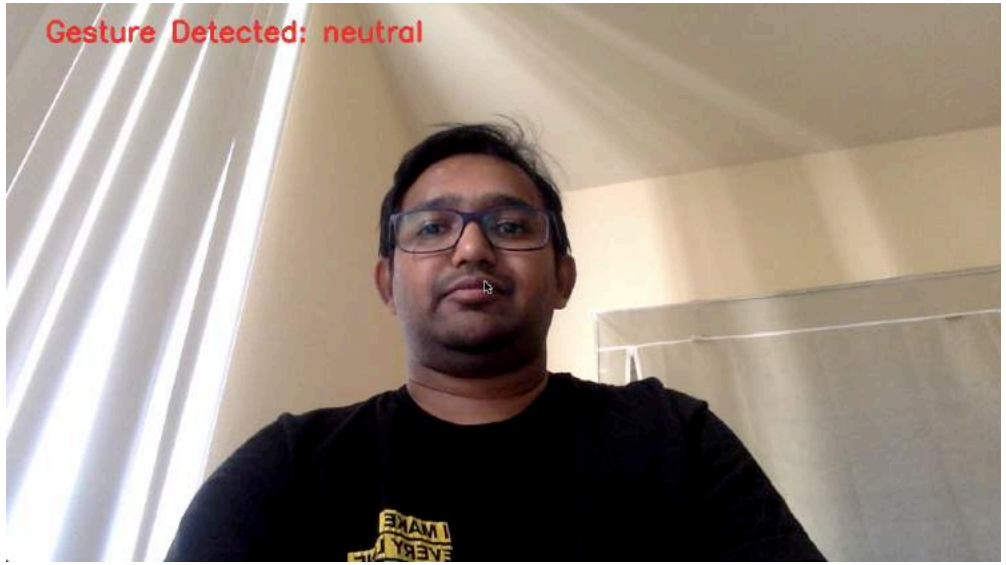


Fig. 14. Face Gesture Detection in action

Head Movement Detection

Head movement detection is divided into two phases:

1. Calibration phase
2. Detection phase

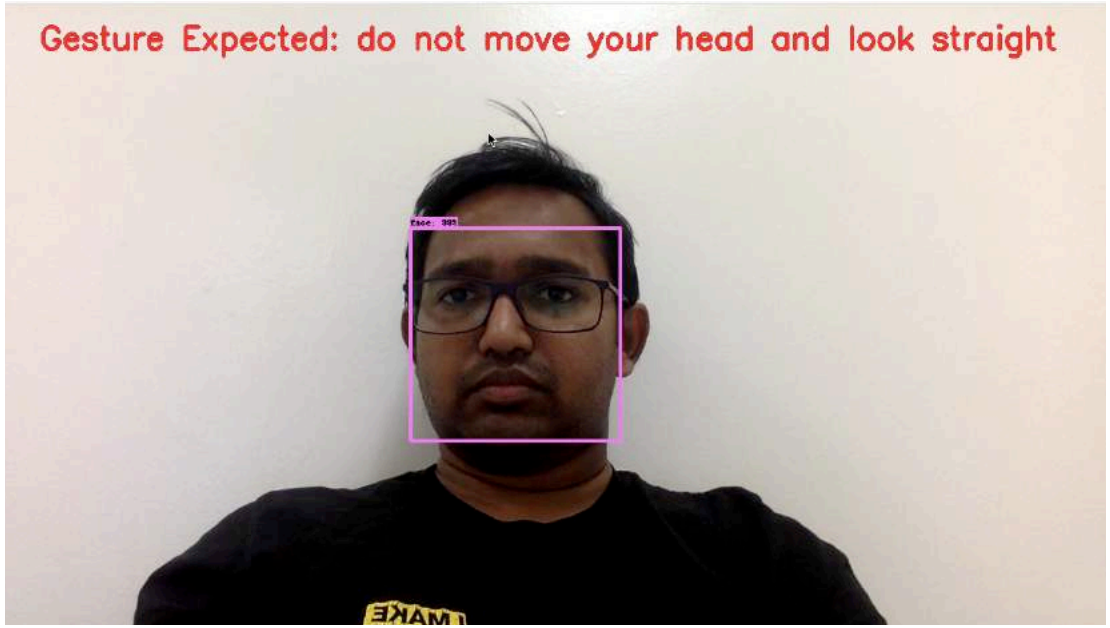


Fig. 15. Head movement detection tool in action

EXPERIMENTS

Design and Architecture and all the sub-components from Implementation are combined together to build the IHFTES.

Output generated using IHFTES is shown in Atom editor and saved in a file name SHFTE (Short for Simple Hands-Free Text Entry) on desktop.

IHFTES starts with calibration phase of head movement, where a users head movements are calibrated.

After calibration phase, user is required to return to the normal position.

User is show the alphabetical layout on return to the normal position.

User could select alphabet-symbols from selecting the correct choices from the layout, or by selecting one of the option.

User is presented with further suggestions based on previous suggestion opted.

Testing and Observation

Sub-components of IHFTES are unit tested separately and after the integration testing is performed once the sub-components are merged.

Based on testing on users and collecting their feedback, following sub-components were further analyzed to make more precise observation:

1. Testing and observation based on detecting facial region
2. Testing and observation on head movement detection

Testing And Observation Based on Detecting Facial Region

1. Error in presence of bright light-source next to user or camera.
2. Error while detecting facial region in the absence of light.

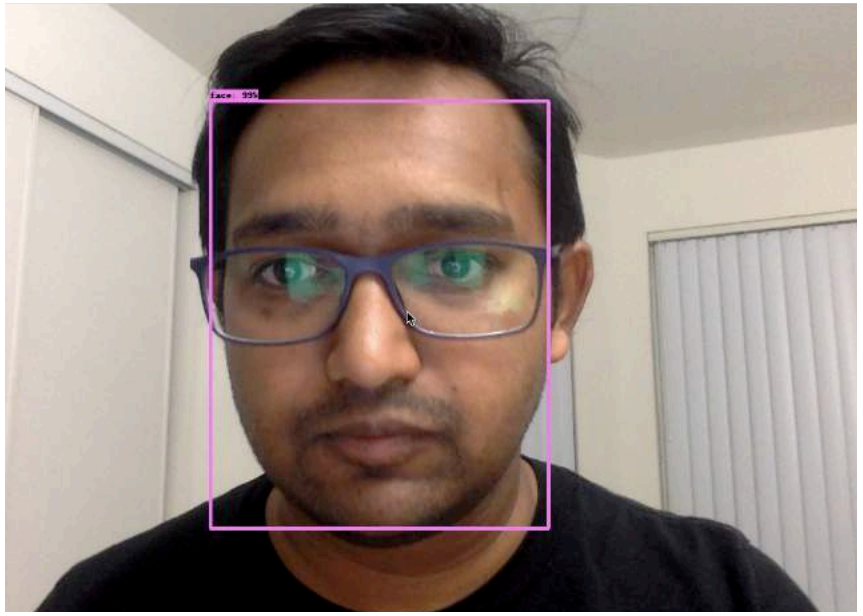


Fig. 16. Effect of presence of light-source in proximity

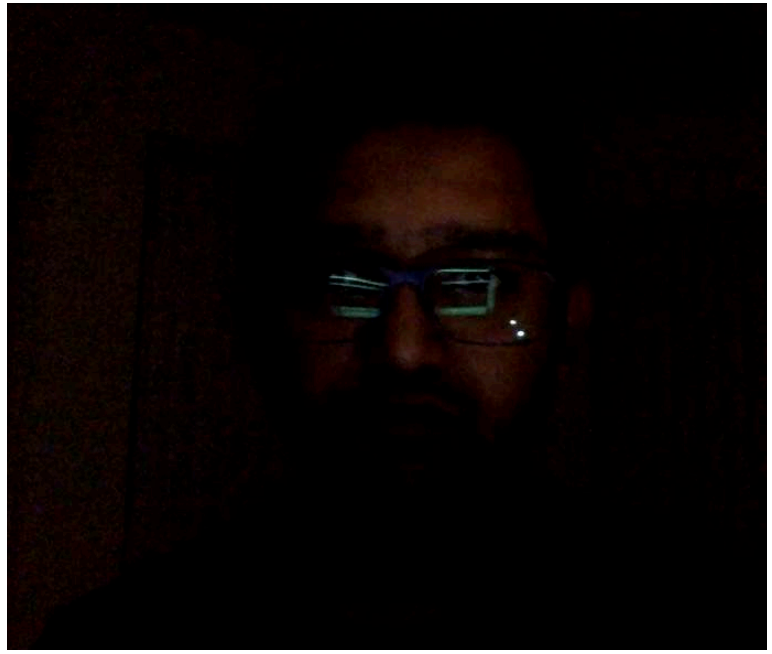


Fig. 17. Effect of absence of light source

Testing And Observation Based on Head Movement

Based on testing head movement detection on users, following feedback was observed:

1. Rapid and continuous head movement in wrong direction, shown in Fig 18.
2. Shifting the head in one of the direction, which makes the instruction on screen invisible to user, shown in Fig 19.
3. Multiple faces in an image frame, shown in Fig. 20.

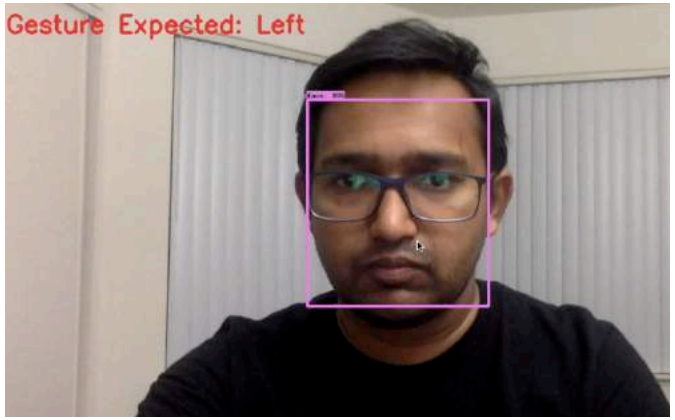


Fig. 18. Rapid Left Movements

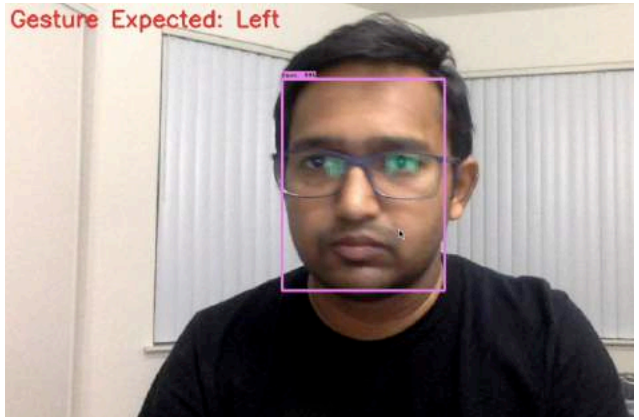


Fig. 19. Extreme Left Movements

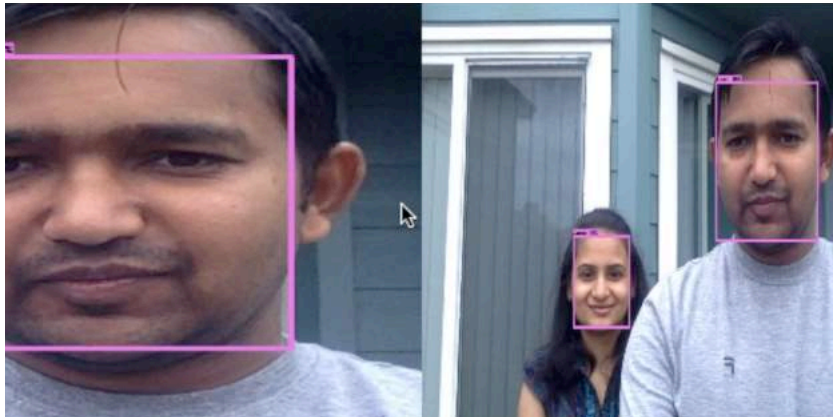


Fig. 20. Detecting primary face in multiple faces

Analysis Based On Configuration of System

TABLE I. Analysis based on the computer configuration

Sr. No.	Device Platform	RAM	GPU	Avg. Frames/Sec
1.	Windows 10	16 GB	6 GB	16
2.	Mac OS	8 GB	N.A.	1.5

Analysis Based On Writing Context Using IHFTES

TABLE II. Analysis for typing words using IHFTES

Dictionary Word	User 1	User 2	User 3
SIN	20 sec	34 sec	65 sec
COP	15 sec	20 Sec	45 sec
MUST	25 sec	33 Sec	45 sec

CONCLUSION AND FUTURE WORK

IHFTES provides an novel way to generate text based characters.

Modal architecture of IHFTES allows to access symbols more swiftly.

Placement of alphabet-symbol based on frequency of characters make it more user friendly.

Suggestion of word based on the proximity to partial word typed is more likely to end up as users choice.

Future Work:

Provide flexibility to send emails using the interface.

Provide flexibility to manage file related operations.

Smart suggestion based on context shall be provided.

REFERENCES

[1] A. Nowosielski, "3-Steps Keyboard: Reduced Interaction Interface for Touchless Typing with Head Movements," in Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017, Polanica Zdroj, Poland, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-59162-9_24

[2] A. Nowosielski, "Two-Letters-Key Keyboard for Predictive Touchless Typing with Head Movements," in Proceedings of International Conference on Computer Analysis of Images and Patterns, Ystad, Sweden, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-64689-3_6

[3] C. Lv, T. Zhang and C. Lin, "Face detection based on skin color and AdaBoost algorithm," in Control And Decision Conference (CCDC), 2017 29th Chinese, Chongqing, China, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7978729/

[4] S. Ma and L. Bai, "A face detection algorithm based on Adaboost and new Haar-Like feature," in 7th IEEE International Conference on Software Engineering and Service Science (ICSESS) , Beijing, China, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7883152/

[5] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23 - 38, 1998. [Online]. Available: https://ieeexplore.ieee.org/document/655647/

REFERENCES

[6] Google, "Tensorflow," [Online]. Available: https://www.tensorflow.org/.

[7] "Tensorflow Detection Model Zoo," [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

[8] "Tensorflow tutorial for Object Detection API," [Online]. Available: https://pythonprogramming.net/introduction-use-tensorflow-object-detection-api- tutorial/.

[9] "Atom editor home page," [Online]. Available: https://atom.io/.

[10] J. Kaufman, "google-10000-english," [Online]. Available: https://github.com/first20hours/google-10000-english.

[11] Z. Zhang, P. Luo, C. C. Loy and X. Tang, "Learning Social Relation Traits from Face Images," in International Conference on Computer Vision (ICCV), Santiago, Chile, 2015. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/socialrelation/index.html.

DEMO

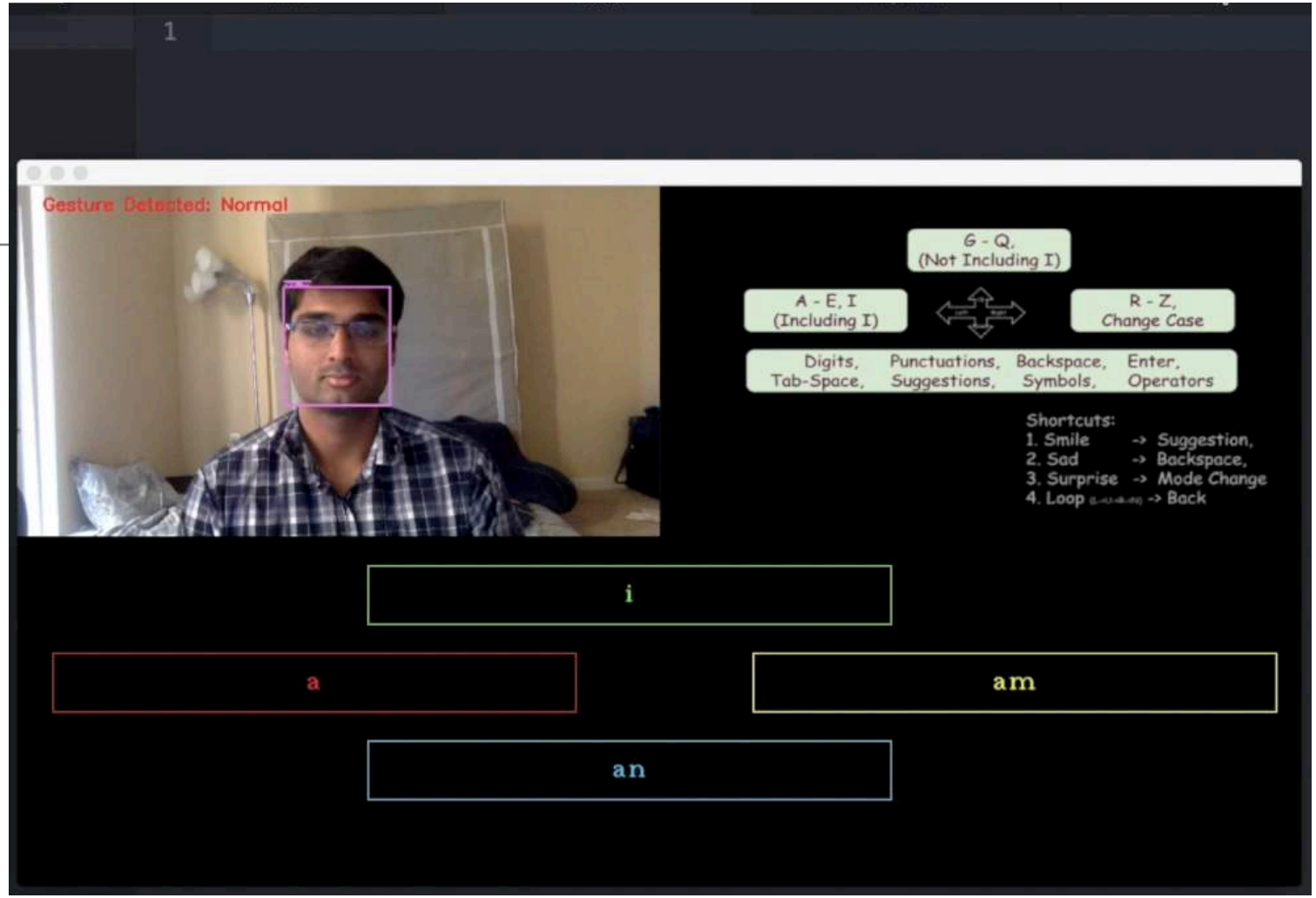


Fig. 21. Working demo of IHFTES