Facial Recognition Smart Cap for Convenient Typing System

Changhyeon Park
sac7160@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Hojeong Lee
leeho@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Yubin Lee
dkch3236@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

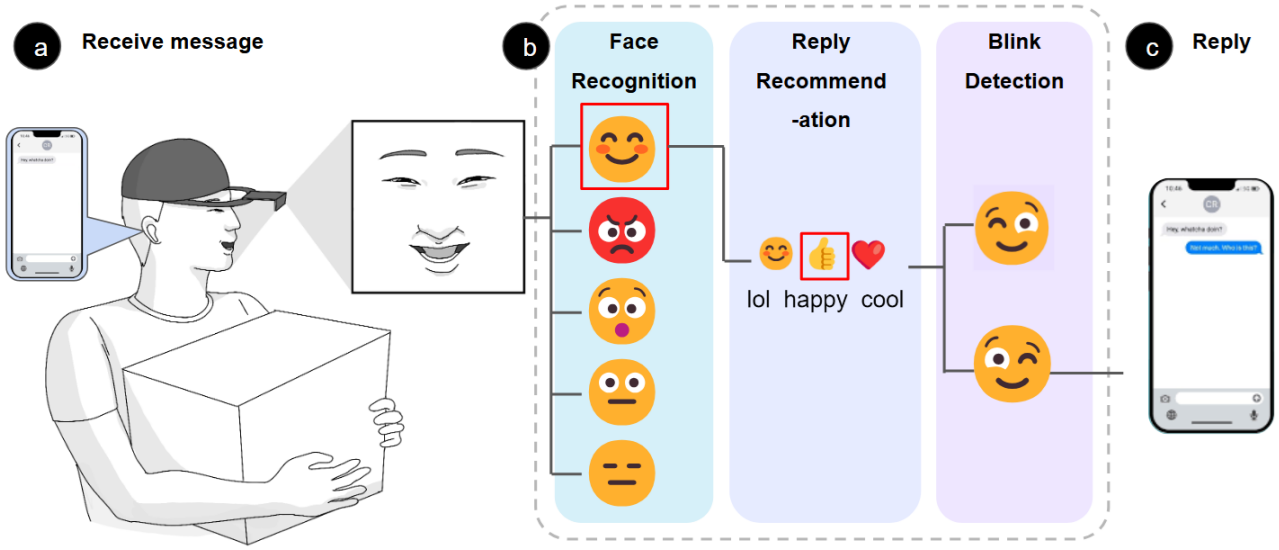

Figure 1: Flow of employing our system for communicating with facial expressions(right).

ACM Reference Format:
Changhyeon Park, Hojeong Lee, and Yubin Lee. 2024. Facial Recognition Smart Cap for Convenient Typing System. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the advancement of extended reality(XR) technology, augmented reality(AR) and mixed reality(MR) wearable devices are entering our daily lives. Compared to traditional mobile communication devices like smartphone and smartwatch, AR and MR devices are another alternative that can deliver the message directly on the users' field of view.[7] Previous research in human-computer interaction has concentrated on refining input mechanisms for AR/MR environments, such as in-air typing[11], gaze tracking[2], and gesture recognition[4]. However, these input systems have a limitation that even when typing a simple reply, the user has to pause their activities and focus on typing or selecting the right emojis. In addition, when conveying emotions, there is a time delay from the moment the user feels a certain emotion to typing out characters or searching for emojis, resulting in extended response times. With AR/ MR input systems, there is a considerable increase in response time because of low accuracy and usability.

To address these challenges, we propose a novel wearable smart-cap that facilitates user-friendly, intuitive input through facial recognition. The system employs a lightweight camera and microcontroller attached to the AR smart-cap to distinguish emotions from the user's facial expressions and offer prompt suggestions for simple messages or emojis, thus improving usability and responsiveness.

We built a prototype by attaching a camera and arduino board to a cap, and collect our own dataset to train a model that can recognize facial expressions and winks of users. In the process, we gained various insights on how to recognize human faces. We learned that the upper face generally plays a more important role than the lower face in the classification of facial expressions, and that different facial parts work as main features depending on the kind of emotion being recognized. We also learned that when using individual datasets, higher train accuracy can be achieved with less training step, but it can be more vulnerable to overfitting.

In our final prototype, we used a general dataset of 2300 images of three people's faces in different lighting conditions. We deployed the model on a wearable prototype to see how it performs in real-world situations. Our final prototype's accuracy is shown as follow; right wink : 0.48, left wink : 0.43, happy : 0.30, angry : 0.23,closed : 0.68, open : 0.31, and it couldn't detect the surprised.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 RELATED WORKS

2.1 AR/MR typing methods

2.1.1 Mid-air hand gesture. Since the keyboard UI in AR/MR environments is usually located in mid-air, mid-air hand gestures were mainly proposed as a typing method for AR/MR environments. Many commercial AR devices, such as Vision pro or hololens, offer a default typing interaction that users touch the keys of a virtual keyboard in the air with their hands. To reduce the arm fatigue and error rate of this typing method, several methods have been proposed that allow typing to be performed with smaller gestures. For instance, ThumbAir[3] has designed an interaction that allows people to type in an AR environment with a single thumb gesture, just as they would type with a smartphone in one hand.

2.1.2 Handheld controller. Like mid-air gestures, the most common way to type on AR devices today is using a handheld controller.Hololens provides a default palm-sized keyboard module that can be connected to the HMD via Bluetooth. With the controller, users can get physical haptic feedback and type outside of the camera tracking range. However, the main drawback is that it requires users to carry a separate controller and hold it in their hands, which limits their hand usage.

2.1.3 Physical touch. TapType[8] has developed a system that allows for text input in AR environments by analyzing the vibration and shape of the hand as it taps a physical surface. Due to the need for a flat surface to tap on, the system can only be used in limited situations, such as sitting at a desk. On the other hand, interactions that utilize the user's skin as the touch surface, such as fingers (DigiTouch[10]), can be used in any situation and have the advantage of being bare-handed. However, they also have the disadvantage of being difficult to learn since the gestures used are unintuitive.

2.1.4 Eye gaze. iText[6] proposed an effective method for AR text typing that utilizes eye gaze routes and blink detection. This interaction is hands-free and socially acceptable since the gestures are not visible to others. However, the continuous rapid eye movement can easily fatigue the user and interfere with their vision in walking scenarios, posing a safety issue.

2.2 Emoji suggestion system using facial detection

Emojis can effectively convey emotions when communicating with others. However, selecting the most appropriate emoji from a variety of emoji can be tricky. To address this issue, various systems have been proposed in the existing literature that recognize human facial expressions and automatically suggest an appropriate emoji. For example, Face2Emoji[1] proposed a system that captures a user's face with a smartphone's front camera, classifies it into different emotion classes, and lists emoji with similar emotions in order of similarity. Another study[5] recommends emoji using the same process, but with improved accuracy of emotion classification by analyzing the user's facial image taken in real-time through a smartphone and heart rate data measured by a wearable sensor device. Users can create customized emoji using Apple's Memoji[9] that reflect their voice and facial expressions. All the systems studied so far require the smartphone's front-facing camera to capture a face. None of these systems have been designed or implemented for hands-free situations or AR environments. Next, we describe our wearable prototype and the training of a machine learning model for single text typing.

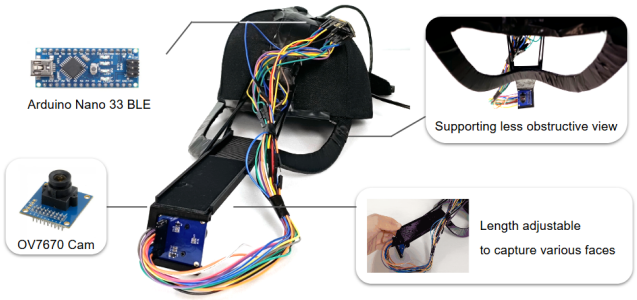

Figure 2: Overall hardware consist of cap, bridge structure, and microcontroller and camera module.

Figure 3: Data collection setup

3 SYSTEM IMPLEMENTATION

3.1 Hardware

We propose a compact smart cap for convenient typing with a system featuring a lightweight microcontroller and a small camera module. We utilized the OV7670 camera module (Arducam) with dimensions of 35.4 x 35.5 cm, paired with an Arduino Nano 33 BLE board serving as the microcontroller for image processing tasks. The camera module could support the resolution of VGA (640 x 480), but we collected the image with QQVGA (128×160) in order to process the data with small microcontroller. In addition, due to the limitations of the camera module's supported focal length, we created a bridge structure to capture the user's entire face and not to obstruct their view at the same time. The structure was designed to be adjustable in length from 10 to 15 cm to capture faces of different sizes and shapes. The camera is fixed at the end of the bridge with tilted at a 35-degree angle allowing for comprehensive capture of

eye, eyebrow, and mouth movements. The microcontroller is affixed to the front area of the cap.

3.2 DataSet

3.2.1 Data collection. Training data and test data were collected from 3 participants. : 1 male 2 female, 27,26,26 years old respectively. All participants were able to perform facial expressions and winks for each emotion. To avoid learning a model that is fitted to a location, each participant acquired data from a different location, and to improve the robustness of the model to lighting, we acquired data under different LED lights as shown in Figure 3. The device was also re-worn after every 30 images to minimize bias due to camera position. The data acquisition time for each participant was about two hours.

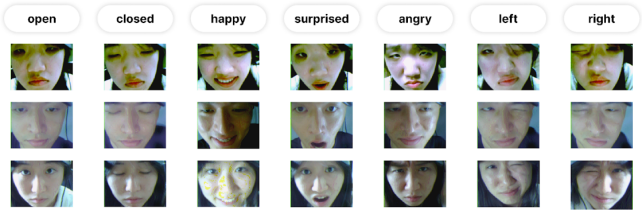

Figure 4: DataSet

3.2.2 DataSet. Data was acquired for a total of 7 categories (angry, happy, surprised, neutral-eyes-closed, neutral-eyes-open, left wink, right wink). For training, 411 angry images, 349 happy images, 255 surprised images, 402 neutral-eye-closed images, 299 neutral-eye-open images, 186 left wink images, and 183 right wink images were obtained. We acquired 30 images for each emotion as data for the test. (Figure 4)

To identify the main facial features that the emotion recognition model mainly utilizes, the same dataset was cropped into an upper face containing the eyes and a lower face containing the mouth.

3.3 Model

Due to the capacity of the Arduino, we had to design a single model to classify winks and facial expressions simultaneously, using a facial image as input. Therefore, our model classifies the facial image into 7 classes (open, closed, left wink, right wink, happy, surprised, and angry). In the use case level, open and closed classes will be recognized as a neutral state. We divide those classes to increase the model's accuracy. Since the model had to be trained on a limited-size dataset that we collected ourselves, we performed transfer learning using a model that was pre-trained on a large image dataset. As a base model for transfer learning, we used MobileNetV2 trained with ImageNet due to its compact model size.

We trained the model under different conditions by varying the dataset used for training (individual/general) and training epoch. As a result, the model trained on the general dataset for 100 epochs had the highest accuracy and we decided to use it for our prototype implementation. We will present the detailed training results in the Result section.

Unfortunately, the size of the final model was too large to deploy on the board. Therefore we once again modified the model

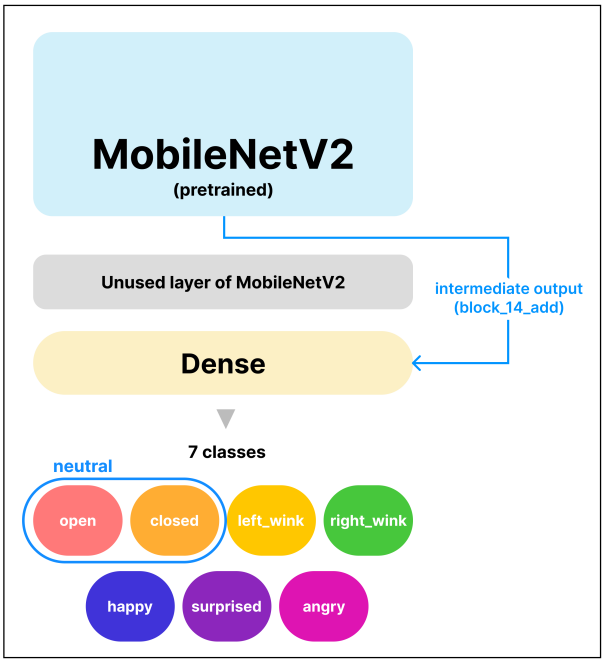

Figure 5: Final model structure used for prototype

structure to reduce the model size. To reduce the number of parameters to largely decrease the model size, we use the intermediate output('block 14 add' layer) of our base model, MobileNetV2 as an input of our final dense layer. Fig 5 describes our final model structure.

3.4 Reply Recommendation System

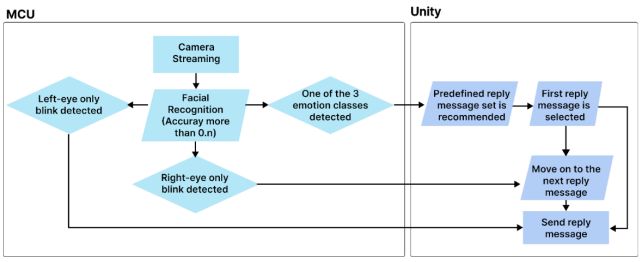

Figure 6: Interaction flow of the usage of simple typing system prototype

To suggest novel texting system, we made a unity AR prototype that could be deployed on user's smartphone. Fig 6 describes the interaction flow of the usage of our system. When a user makes a specific facial expression and the probability of that class is higher than 0.50, arduino IDE send the string data of the specific emotion class through serial communication between Unity 3D. As unity receive the string data of emotion class, it trigger the reply recommendation system showing related emoji or short reply from the

predefined dataset. The first reply is marked to be chosen when the recommendation UI first appear. Then, if the right blink is detected, the choice mark move on to the next reply. If the left blink is detected, the selected reply is finally sent to the partner.

4 RESULTS

4.1 Model training

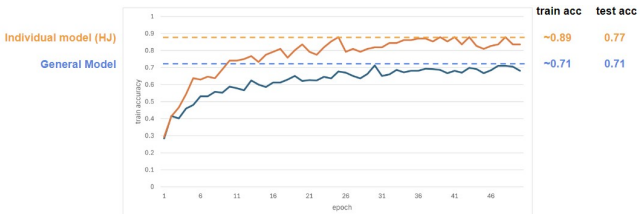

Figure 7: Leaning curve(train accuracy) and test accuracy of Individual(HJ) / General model

<Individual Model>

Method	Data	Test accuracy (general)	Test accuracy (individual)	epoch
Individual	HJ	0.495238095	0.771428571	50
Individual	CH	0.333333333	0.657142857	50
Individual	YB	0.304761905	0.571428571	50
<General Model>				
general	HJ+CH+YB	0.714285714		50
general	HJ+CH+YB	0.742857143		100

Figure 8: Test accuracy based on different train dataset types; Individual, General

4.1.1 General Dataset VS Individual Dataset. Since our final dataset consisted of images from 3 different people, we created 3 individual datasets by categorizing the images for each person, and one general dataset containing images from everyone. We compare the model training results using each dataset.

The results are as follows (Figure 7, Figure 8). The individual model showed better train accuracy than the general model. However, the test accuracy of the individual model decreased significantly compared to the train accuracy when the general model's train accuracy and test accuracy were not significantly different. It shows that the individual model has the advantage of learning faster than the general model, but is vulnerable to overfitting.

The individual models often showed higher test accuracy than the general model, when the test dataset was the individual dataset they trained on. All of them showed severely lower accuracy when the test set was the general dataset. We decided to use the general model to implement the prototype for two reasons. First, the test accuracy of the general model is not that much smaller than the maximum test accuracy of the individual model. Second,we don't know how the environment when we test the prototype will be different from the data collection environment, and the general model is expected to be more stable to environmental changes. After making the decision, we increased the number of train epochs from 50 to 100 to improve the accuracy of the final model, and we were able to get a slightly higher test accuracy.

4.1.2 Model Parameter Reduction. As mentioned in the 'Model' section, the final model was too large to deploy. Therefore we reduced the number of parameters in the base model and retrained it under the same conditions as the previous final model. The test accuracy of the final model is 0.55.

4.2 Emotion Inference Facial Features

4.2.1 Full Face VS Upper Face VS Lower Face. We assumed that just as humans use different facial features to recognize different emotions, machine learning models would do the same. Therefore, we compared the accuracy of the model trained with full face image, upper face image, and lower face image to check the facial features that the model mainly uses to recognize emotions. The results of the individual model test showed the accuracy of full face : 0.89, upper face : 0.825, lower face : 0.4, and the results of the general model were (upper face : 0.596, lower face : 0.525). Comparing the accuracy of Upper Face and Lower Face, (0.825 > 0.4, 0.596 > 0.525) Upper Face was a more important feature for emotion recognition than Lower Face.

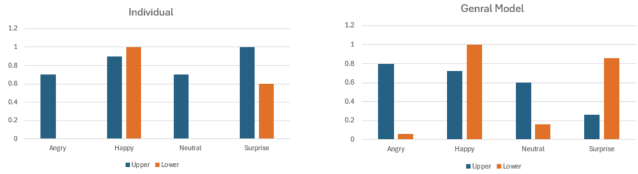

Figure 9: Model accuracy based on facial features by emotion

4.2.2 Main features by emotions. We assumed that the facial features that the models mainly use for each emotion would be different, as the degree of change in different parts of the face varies by emotion, and we compared the accuracy of the models trained with the upper and lower face images to identify the facial features for each emotion. The results are as follows (Figure 9), comparing the results of training only the upper and lower facial images, we could identify the most important features for each emotion. For 'Angry', in both individual and general models, the upper face is the more important feature, so we can see that there is a big difference in the eye area. For 'Happy', the opposite result was found. For 'surprised', the individual and general models showed different results, which we believe is due to bias in data due to human characteristics and facial expression habits.

4.3 Arduino Deployment

We deployed the final model to the microcontroller. However, despite the reduction of the size of our model, it still exceeded the capacity of the microcontroller. In order to succesfully deploy the model to arduino, we reduced the size of the memory buffer which could affect on the performance of the model via the microcontroller, to reach its maximum within the capacity. However, because of low memory buffer and low image resolution, accuracy usually lied around 0.30 and the malfunction commonly occurred. The results of the model through the arduino showed the accuracy of right wink : 0.48, left wink : 0.43, happy : 0.30, angry : 0.23, neutral closed : 0.68, neutral open : 0.31, and it couldn't detect the surprised.

Also, Happy and angry were often misperceived as neutral closed, and neutral open were often misperceived with neutral closed and happy. Because of the frequent error and malfunction of the model using microcontroller, it was unable to successfully test the system through the AR prototype implemented by Unity.

5 DISCUSSION

Although several works developed simple and hand-free texting systems including in-air typting [11], gaze tracking[2], and so on, they were not suitable for the situation where users cannot use their hands to type or where they cannot put much effort and time for texting. To address this problems, we came up with a novel wearable device that could support easy and simple texting through the identification of the emotion.

There are some limitations that could be address in future work. First, our system utilizes microcontroller as Arduino Nano BLE 33 which supports 1 MB of flash memory and 256 KB of SRAM. We found out that this capacity is insufficient to support our model for image recognition with various classes. For the future work, we can utilize the boards with bigger capacities like ESP 32- EYE, which supports the flash memory with 8 MB and 520 KB SRAM, 4 MB PSRAM. We didn't utilize it for current work as it require to use different platform for programming (i.e. ESP IDE). In addition, our device has the limitation of being less socially acceptable due to its low comfort and wearability. Our current camera, OV7670 not only consumes large capacity but also supports focal length which is not suitable for our work which needs to collect the full face as well as regarding the wearability. We can utilize the camera with fish-eye lens to position the camera close to the face so that we can remove the bridge structure and provide better wearability. Lastly, we found out that the complexity of human facial expressions requires a much larger dataset. For the higher accuracy, it is needed to not only increase the performance of the model itself, but also increase the number of dataset.

6 CONCLUSION

We created a wearable hat that allows users to send simple messages using only facial expressions when they can't use both hands. The user can select and send simple messages recommended by facial expressions. For this purpose, we acquired training and test datasets for the emotion recognition model from three participants and deployed it on an Arduino board after training. In addition, to find the main facial features used for emotion recognition, we measured the accuracy by dividing them into upper face and lower face, and analyzed the accuracy of facial features by emotion to find the main facial features by emotion.

Although the bulkiness of the prototype is a limitation, we acquired data under various conditions to reduce bias and maximize robustness, and analyzed key facial features that can help improve the performance of future models and reduce the size of the device. We expect to use this to develop a device with commercially feasible accuracy and size in the future.

REFERENCES

[1] Abdallah El Ali, Torben Wallbaum, Merlin Wasmann, Wilko Heuten, and Susanne CJ Boll. 2017. Face2Emoji: Using Facial Emotional Expressions to Filter Emojis *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 1577–1584. https://doi.org/10.1145/3027063.3053086

[2] Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos H Morimoto, and Margrit Betke. 2021. HGaze Typing: Head-Gesture Assisted Gaze Typing. In *ACM Symposium on Eye Tracking Research and Applications* (Virtual Event, Germany) *(ETRA '21 Full Papers)*. Association for Computing Machinery, New York, NY, USA, Article 11, 11 pages. https://doi.org/10.1145/3448017.3457379

[3] Hyunjae Gil and Ian Oakley. 2023. ThumbAir: In-Air Typing for Head Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 164 (jan 2023), 30 pages. https://doi.org/10.1145/3569474

[4] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300244

[5] Jesung Kim, Mincheol Kang, Bohun Seo, Jeongkyu Hong, and Soontae Kim. 2023. Effective Emoticon Suggestion Technique Based on Active Emotional Input Using Facial Expressions and Heart Rate Signals. *Sensors* 23, 9 (2023). https://doi.org/10.3390/s23094460

[6] Xueshi Lu, Difeng Yu, Hai-Ning Liang, and Jorge Goncalves. 2021. iText: Hands-free Text Entry on an Imaginary Keyboard for Augmented Reality Systems. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 815–825. https://doi.org/10.1145/3472749.3474788

[7] Rufat Rzayev, Susanne Korbely, Milena Maul, Alina Schark, Valentin Schwind, and Niels Henze. 2020. Effects of Position and Alignment of Notifications on AR Glasses during Social Interaction. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society* (Tallinn, Estonia) *(NordiCHI '20)*. Association for Computing Machinery, New York, NY, USA, Article 30, 11 pages. https://doi.org/10.1145/3419249.3420095

[8] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 497, 16 pages. https://doi.org/10.1145/3491102.3501878

[9] Apple Support. 2024. *Use Memoji on your iPhone or iPad Pro*. https://support.apple.com/en-us/111115

[10] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (sep 2017), 21 pages. https://doi.org/10.1145/3130978

[11] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 539–548. https://doi.org/10.1145/2807442.2807504