Proceedings of the ASME 2020
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2020
August 17-19, 2020, Virtual, Online

DETC2020-22365

INTEGRATED AND CONFIGURABLE VOICE ACTIVATION AND SPEAKER VERIFICATION SYSTEM FOR A ROBOTIC EXOSKELETON GLOVE

Yunfei Guo
Robotics and Mechatronics Lab
Department of Electrical and
Computer Engineering
Virginia Tech,
Blacksburg, Virginia 24061

Wenda Xu, Sarthak Pradhan
Robotics and Mechatronics Lab
Department of Mechanical Engineering
Virginia Tech,
Blacksburg, Virginia 24061

Cesar Bravo
Carilion Clinic Institute of
Orthopaedics and Neurosciences
Virginia Tech Carilion School of Medicine
Roanoke, VA 24014

Pinhas Ben-Tzvi *
Robotics and Mechatronics Lab
Department of Mechanical Engineering
Virginia Tech,
Blacksburg, Virginia 24061

ABSTRACT

Efficient human-machine interface (HMI) for exoskeletons remains an active research topic, where sample methods have been proposed including using computer vision, EEG (electroencephalogram), and voice recognition. However, some of these methods lack sufficient accuracy, security, and portability. This paper proposes a HMI referred as integrated trigger-word configurable voice activation and speaker verification system (CVASV). The CVASV system is designed for embedded systems with limited computing power that can be applied to any exoskeleton platform. The CVASV system consists of two main sections, including an API based voice activation section and a deep learning based text-independent voice verification section. These two sections are combined into a system that allows the user to configure the activation trigger-word and verify the user's command in real-time.

1 INTRODUCTION

According to statistical data published in 2010, over 6.7 million of U.S. adults have difficulty to grasp or handle small objects [1]. The RML exoskeleton glove (RML glove) is designed to be used as an assistive and rehabilitation device for Activities of Daily Living (ADL) [2].

The mechanical design of the RML glove has improved over the years and the current mechanical design is the fifth generation. Some of the previous RML glove designs are shown in Fig.1. Previous RML glove designs have used motors with cable transmission [3], pneumatic actuators [1], and serial elastic actuators [2].

The current generation RML exoskeleton glove is based on a rigid linkage mechanism using a series elastic actuator (SEA) on each finger to control each of the linkage mechanism per finger. Each finger can perform 3 degrees of freedom (DOF) motion and exert 35N force through the linkage mechanism [2]. The RML glove can perform 8 grasps simplified from the HUST dataset [4]. There are 33 different grasps in the HUST dataset [5] that have been narrowed down into the following 8 basic grasps: Cylindrical Wrap, Sphere, Two Finger Pinch, Prismatic Small Stick, Index Extension Wrap, Disk, and Flat Parallel [4]. The mechanical design of the RML glove has significantly improved; however, the control of the RML glove remains to be desired. The current generation RML glove can only be controlled by switches, buttons, and keyboards. Users with limb and hand disabilities will not be able to use buttons and switches to control

*Corresponding author – bentzvi@vt.edu

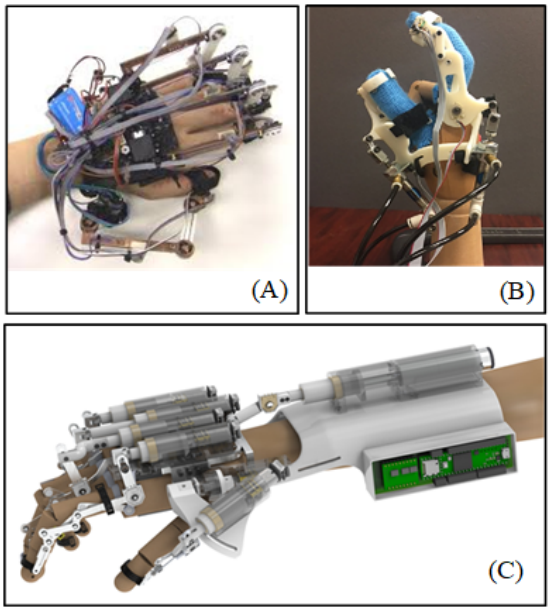

Figure 1. (A) RML glove with tendon actuation. (B) RML glove with pneumatic actuation. (C) RML glove with SEA actuation

the exoskeleton.

In order to control RML glove, the command system needs to be convenient, accurate, secure, and low computational cost. A high efficient user intention detection interface needs to be designed to control exoskeletons glove system.

To fulfil the aforementioned requirements, researchers have previously designed three different types of user-interfaces for the exoskeleton device. These different methods require minimal limber movements. All these user-interfaces are more superior than using buttons and switches.

L. Randazzo, et al. [6] proposed an electroencephalogram (EEG) based intention detector. The usage of EEG can detect the user's intention within 400ms and provides an accuracy above 70%. The advantage of using a brain-computer interface is that the user can naturally control the exoskeleton with moderate latency. However, using EEG requires the user to wear a large EGG device, and the accuracy (above 70%) of EEG-based detection is relatively low. It is not portable and the accuracy does not fulfill our requirement.

Daekyum Kim, et al. [7] proposed a vision-based intention detector. The user needs to wear glasses with a camera; the camera shares the same view of the user and uses it to detect their intention. This system also has a moderate response speed of 300-800ms. However, multiple objects in the camera or occlusions between objects will affect the system's overall performance.

Xuefeng Wang, et al. [8] proposed using Apple's Siri to control their exoskeleton glove, with the program running on an iPhone. According to Apple, Siri has a relatively low (5%) word error rate and is able to run without noticeable latency on an iPhone. It is proved to be robust and stable by many Siri users. The lack of configurable activation keyword is inconvenient when using the glove in public. The lack of speaker verification can also cause security related problems.

Compared with EEG and vision detection methods, voice detection does not have major drawbacks. The accuracy is relatively high and the latency is moderate. Using voice control will be the most applicable and appropriate method to control RML glove.

To design a portable voice command system for the RML glove, the activation trigger-word needs to be configurable and all commands need to be verified on a platform with limited computing power. With the help of application programming interface (API) from large companies such as Google and Apple, low computational cost configurable voice activation and recognition can be achieved. There is no available API for text-independent speaker verification. Nagran, et al. [9] used a deep neural network to accomplish text-independent speaker verification and received a 2.87% error rate. Since VGG-M has less layers and parameters, it is faster than ThinResNet in terms of computational speed. However, VGG-M is not the fastest network that can be used on portable devices. In this paper, MobileNetV1 was used to build a text-independent speaker verification on a platform with limited computing power. VGG-M will also be tested and used as the baseline method to compare the performance gain.

2 RELATED WORK

2.1 VGG-M Speaker Verification

VGG-M with Softmax loss function is a deep learning approach to the voice verification system proposed by Nargrani, et al. [9]. This deep learning voice verification system achieves better accuracy (10.2%) than the non-deep learning baseline (15.0%).

The voice verification system consists of three sections. The first section is prepossessing, where each audio file is divided into several 3-second audio clips. Each audio clip is turned into a 512x300 spectrum using Fourier transform. The spectrum is treated as 2D images and feeds into an utterance level feature extractor.

The second section consists of utterance level feature extraction, where 1251 speakers will be classified as 1251 classes by deep neural network and the last layer will be removed. The remaining network will be used as a feature extractor. The VGG-M network is modified based on VGG16 [9]. Compared with VGG16, VGG-M can take various lengths of 2D spectrum as an input. The output of the feature extractor will be used in the verification process.

The third section consists of verification calculation. If the cosine distance between two audio samples are within the thresh-

old, these two samples will be considered as the same class. The baseline method changes the average pooling layer to match the test-time length, so that the network can take various length input. To make the verification more robust, the authors also proposed Test Time Augmentation 2 and Test Time Augmentation 3 method, which randomly select 10 samples from the samples and calculates the average distance of features [9]. The VGG-M network with Softmax loss, Global Average Pooling, and Test Time Augmentation 2 will be used as the deep learning baseline in this research.

2.2 MobileNet

When using personalized voice activation and command system on exoskeletons, the computing speed is crucial. The networks proposed by Nagran et al. [9] are not the fastest network to run on a mobile device. Andrew et al. [10] proposed an efficient convolution neural network used on image classification. MobileNet-224 (MobileNetV1) achieves similar accuracy (70.6%) as VGG16 (71.5%) on ImageNet dataset. MobileNet-224 has far less parameters (4.2 million) compared to VGG16 (138 million), thus, it is faster than VGG16. The VGG-M network is modified based on VGG16, which has similar accuracy in image classification.

3 PROPOSED APPROACH

3.1 Hardware System Overview

The hardware system shown in Fig.2 contains a smartphone, a microphone, and a micro-controller. The microphone input raw data to the smartphone and all the personalized voice activation and verification can be calculated on-board. The micro-controller will be placed on the exoskeleton to receive verified commands. The smartphone and micro-controller communicate through Bluetooth. With minimal carry on devices and cable connection, this hardware system is designed to maximize portability and convenience of use.

3.2 Software System Overview

The configurable personalized voice activation and command system can be divided into two sections. The first section controls the activation process and the second section controls the verification process. Fig.3 shows the thread assignments and the tasks that occur in each thread. The configurable voice activation section begins with the raw audio data followed by outputting the accepted audio data. The microphone streaming callback will continuously generate 0.5-second audio segments. The audio collection thread contains a noise reduction filter and a loudness filter. If the loudness is greater than the threshold and a complete sentence has been detected, then the audio collection thread will enter the pre-active mode. The audio data queue will send data from the audio collection thread to the audio

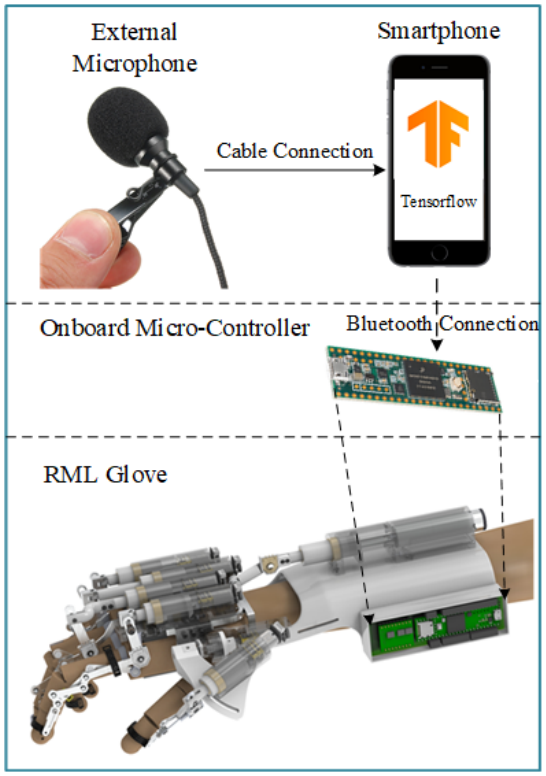

Figure 2. Hardware System and Peripheral Device Connections

processing thread under pre-active mode. The voice processing thread consists of two stages. The command detection stage uses voice recognition API. After the command is accepted, it enters the speaker verification stage. The MobileNetV1 speaker verification system will verify if the audio belongs to the enrolled speakers.

3.3 Configurable Voice Activation

The voice activation system includes a noise reduction filter, a loudness filter, and a command detector. The system is designed to detect possible activation commands using minimal computational cost. The system will enter a pre-activated mode if the intensity of the human voice is greater then a threshold. A human-like voice will be checked using voice activation API to verify if an activation trigger-word is present. If the activation trigger-word is present, the system will enter into an activated mode and the audio will be passed to the voice verification system.

The raw input of the microphone voice contains high and low-frequency noise. With a noise reduction filter, the system will enter the pre-activated mode less frequently on false positive activation in a noisy environment.

The noise reduction filter is used to minimize high and low-

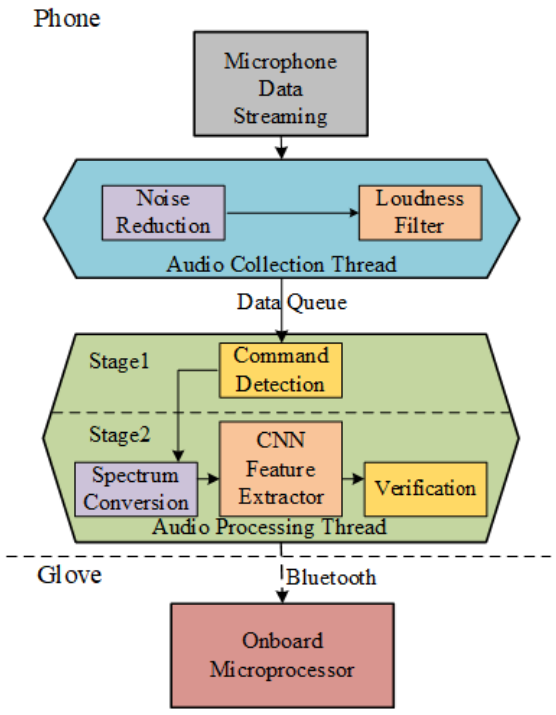

Figure 3. Task Diagram of Configurable Personalized Voice Activation and Command System

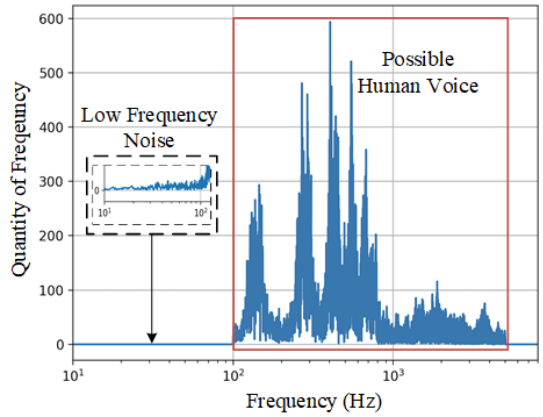

Figure 4. Fourier Transform Result of the Original and the Filtered Audio Data

frequency noise. It converts raw input from the microphone to time-independent frequency domain using Fourier transform. The total quantity of each frequency is measured. High and low-frequency noise is trimmed. The time-independent Fourier transform [11] of the original audio data and the filtered audio data is shown in Fig. 4. The remaining frequencies contain potential human voices, which will enter the loudness detector.

The loudness detector will aggregate small audio segments into complete sentences. Audio data that has a loudness level above a certain threshold will set the system to pre-active mode. The input data is a 0.5-second filtered audio segment. These audio segments will be grouped into a larger audio segment based on the margin between each word. We assume that if the margin between words is larger than 0.5-seconds, the sentence ends. Using this assumption, the loudness detector with audio aggregation is implemented based on the flowchart described in Fig.5.

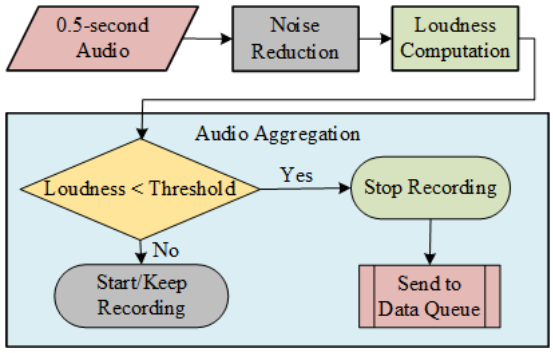

Figure 5. Flow Chart of Loudness Detector with Audio Aggregation

After the loudness detector detects a complete sentence, the audio data is sent to audio processing thread for voice recognition. The loudness detector keeps monitoring the environment. In the voice processing thread, the complete sentence will be recognized using API. Voice recognition API from Amazon, Google, or Apple are proved to be accurate and require low computational cost.

If the system is not activated, the result of the API will be used for activation of the trigger-word detection. If the system is activated, the result of the API will be used as an unverified command. Words that have similar pronunciation with the activation trigger-word or command will be accepted. After checking that the trigger-word or command is valid, speaker verification will be performed using the filtered audio data.

Voice Verification

The voice verification method was improved based on Nagrani et al.'s [9] VGG-M with the Softmax loss function speaker verification method. Instead of using VGG-M, the low computational cost network MobileNetV1 was used. The size of the MobileNetV1 was modified. The size of the network is reduced to 75% of the original network. So, alpha value equals 0.75. The network is modified to adapt 2D spectrum input.

The training procedure is similar to the baseline VGG-M method. Filtered audio data from the previous speaker activation section is converted into a 2D spectrum [11] that has both frequency and time information. The 2D spectrum will be fed into MobileNetV1. The speaker utterance is grouped using the Global Average Pooling (GAP) into 768 features. These 768

features will be classified using a dense layer. The loss function used for classification is the standard Softmax loss function shown in Eq.1, where x is the spectrum input, j is the label of a certain speaker, $X$is the features output from the neural network, k is the number of classes, and w is the weighting vector. The training process is shown in Fig 6.

$$P(y=j|x) = \frac{e^{X^T W_j}}{\sum_{k=1}^{K} e^{X^T W_k}} \tag{1}$$

During the speaker verification procedure, the last dense layer of the MobileNetV1 is detached. The output of the network has 768 features. The distance between two audio features vector A and B is calculated. If the distance is less than an enrollment threshold, these two audios are from the same speaker. The verification procedure is shown in Fig. 6. The distance between features is calculated by cosine distance [12]. The cosine distance (D_c) is defined by Eq. 2.

$$D_c = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2}$$

During the enrollment process, the user needs to record a x-second long enrollment audio and then the audio is randomly cut into n audio segments each having a 3-second length. These n audio clips will be processed by the neural network and each audio clip will be turned into a feature vector (F). The average enroll-feature vector (F_{avg_enroll}) can be calculated based on Eq. 3.

$$F_{avg} = \frac{\sum_{k=1}^{n} F_n}{n} \tag{3}$$

During the verification process, the input test audio will be cut into n audio segments each having a 3-second length. The average test-feature vector (F_{avg_test}) can be calculated based on Eq. 3.

To calculate the verification threshold, the average distance between the same speaker and different speakers over a verification training dataset can be calculated. The verification training dataset contains K speakers. The threshold (T) is chosen by calculating the mean distance between the average enroll-feature (F_{avg_enroll}) and the average test-feature (F_{avg_test}) over K speakers as shown in Eq. 4.

$$T = \frac{\sum_{k=1}^{K} dist(F_{avg\_enroll}, F_{avg\_test})}{K} \tag{4}$$

4 EXPERIMENTS

4.1 Neutral Network Feature Extractor

VoxCeleb1 [13] dataset is used for training the voice verification system in this research. The VoxCeleb1 dataset contains 1251 celebrities giving presentation under different environments. There are over 100,000 utterances and 40 speakers are held out for the verification test. This dataset contains noisy data that is suitable for training a robust speaker verification model.

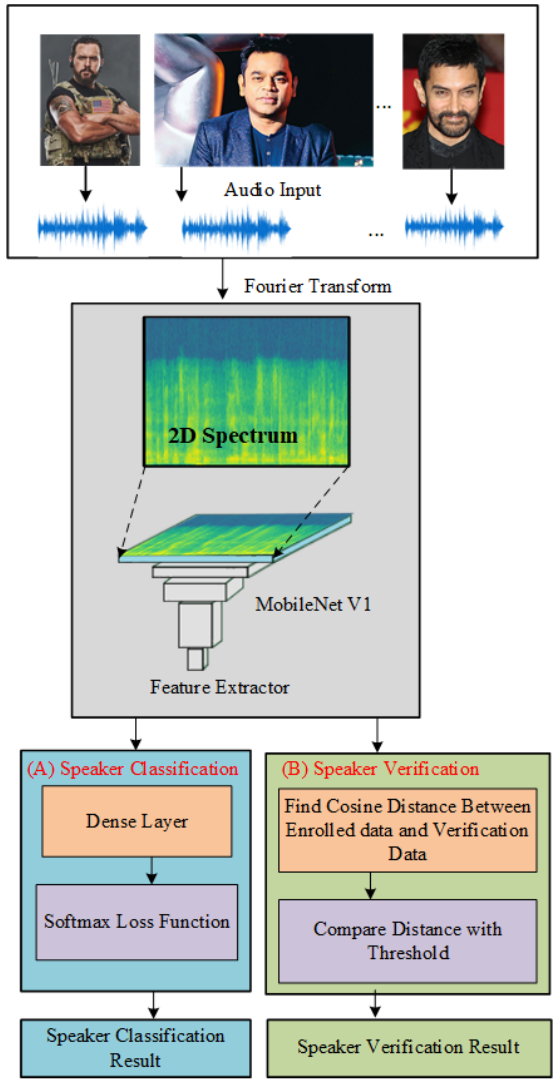

Figure 6. Training and Testing with Modified MobileNetV1. (A) Training Add-on. (B) Testing Add-on

The feature extractor training is performed on VoxCeleb1 dataset using MobileNetV1 and VGG-M. VGG-M will be used as the performance baseline. A 3 seconds audio clip is randomly extracted from each audio file and converted to a 512x300 spectrum using Fourier transform. The 512x300 spectrum is used to train both MobileNetV1 and VGG-M networks. All networks will classify input audio data into 1251 classes.

The MobileNetV1 feature extractor (classification) was trained on VoxCeleb1 dataset with 12 epochs. The training logs are shown in Fig. 7.

The speaker verification dataset is from the VoxCeleb1 ver-

ification dataset. It contains 40 speakers that have IDs from 270-309. During the verification testing process, two audios are given: one is used as enrolled data, and the other is used as test data. The verification test data is randomly divided into 3 segments each consisting of a 3-second audio. The enrolled data is randomly divided into 3 segments each consisting of 3-second audio. The time cost to process 1 clip of a 3-second audio will be measured to compare the performance.

4.2 CVASV System

A timing analysis was performed to each part of the system. The system contains five major parts: reduction filter, loudness filter, voice recognition API, spectrum conversion, and neural network feature extractor. The noise reduction filter and loudness filter are connected in series. To achieve real-time operation, these two filters need to run once every 0.5s. The spectrum conversion, neural network feature extractor, and voice recognition API are connected in series. To achieve real-time operation, this part needs to run once every 3 seconds.

The CVASV system was tested on a Intel(R) Xeon(R) E5-1650 @ 3.2Ghz CPU with no GPU involved in the calculation. The voice recognition API used in the experiment was Google voice recognition API. The neural network feature extractor used was a modified MobileNetV1.

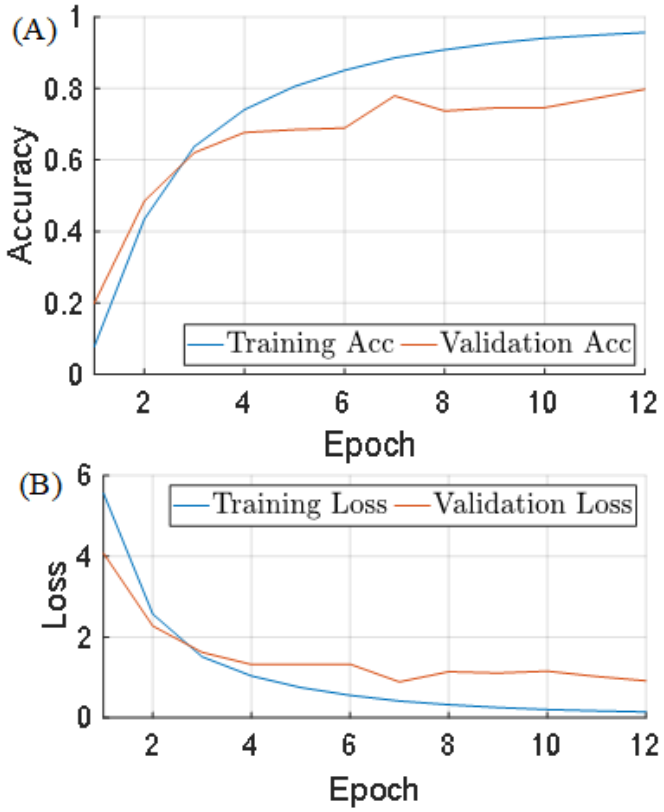

Figure 7. (A) Modified MobileNetV1 Training and Validation Accuracy; (B) Modified MobileNetV1 Training and Validation Loss

Table 1. Comparison between Modified MobileNetV1 and VGG-M

	MBN	VGG-M
Classification Accuracy	**79.8%**	61.2%
Verification Accuracy(D_c)	88.9%	**90.3%**
Time cost	**73.2ms**	88.9ms
Total Parameters	**1,832,544**	17,691,328

Bold text highlights better performance. D_c: Cosine distance.

Table 2. Time Cost to process a 3-second audio clip

	Time Cost	Timing requirement
Noise Reduction Filter	1.34ms	
Loudness Filter(D_e)	1.88ms	
Audio Collection Subsystem	3.22ms	≤500ms
Spectrum Conversion	7.21ms	
Voice Recognition API	34.18ms	
MobileNetV1 Feature extractor	73.2ms	
Voice Recognition Subsystem	114.59ms	≤ 3000ms

5 RESULTS

5.1 Neutral Network Feature Extractor

The performance comparison between the modified MobileNetV1 (MBN) and the VGG-M is shown in Tab.1. The modified MobileNetV1 achieves 18.6% higher accuracy in classification. The modified MobileNetV1 has 1.4% lower accuracy than VGG-M in the verification section. The use of MobileNetV1 as feature extractor was 17.7% faster than using VGG-M.

5.2 CVASV System

The time cost for processing a 3-second audio clip is shown in Tab.2. The time cost for the Audio Collection Subsystem and the Voice Recognition Subsystem are far less than the requirement. No GPU was used in this system. However, further experiments are needed to test on a mobile device to make sure that the system is fast enough to be operated on ARM-based processors without GPU support.

6 CONCLUSION AND FUTURE WORK

This paper proposed a fast, portable, and secure method to control the RML glove through voice activation and proved that the MobileNetV1 can be used as feature extractor for this system. MobileNetV1 achieved 1.4% lower accuracy than VGG-M in verification, but was 17.7% faster. Based on current experiments, the personalized voice activation and command system fulfilled all the requirements for real-time operation on an In-

tel(R) Xeon(R) E5-1650 @ 3.2Ghz CPU. This system will be further tested with an ARM-based processor.

There are some improvements that can be pursued in the future. MobileNetV1 was modified based on the baseline VGG-M method with Softmax loss function. Nangrani et al [9] has proved that using large margin Softmax loss function [14], and adding NetVALD [15] or GhostVALD [16] will improve the accuracy of VGG-M. Adding These features might further improve the accuracy of MobileNetV1. Extending the dataset to VoxCeleb2 [17] will improve the accuracy without any detrimental effects.

The verification section in this paper used cosine distance function to verify different speakers. Using a GMM (Gaussian mixture model) [18] or module-based KNN (K nearest neighbor search) [19] to replace the distance threshold method might further improve the accuracy. Using MobileNetV2 [20] might also improve the computational speed.

When applying the proposed system, a potential challenge may be the ability to differentiate between similar voices. When the input voice command is short, it may affect the detection accuracy. Building a GMM model for each user or using KNN classification might improve the detection accuracy, but it may compromise the computational speed. Commercially available voice recognition APIs might not achieve their claimed accuracy. As such, low accuracy of the voice recognition API may affect the accuracy of the system. Therefore, it may be necessary to create a voice recognition API especially tuned for our system.

The CVASV system will be further tested on the integrated RML glove system with an embedded miro-controller and a smartphone. The personalized voice activation system will also be tested with patients to improve the user interface.

7 ACKNOWLEDGMENT

The authors would like to thank Yi Han, Yunhui Zhu, and Hailin Ren who provided valuable advice and greatly assisted in this research. The authors would also like to gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU.

Research reported in this publication was supported by the Eunice Kennedy Shriver National Institute of Child Health & Human Development of the National Institutes of Health under Award Number R21HD095027. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

[1] Refour, E. M., Sebastian, B., Chauhan, R. J., and Ben-Tzvi, P., 2019. "A General Purpose Robotic Hand Exoskeleton With Series Elastic Actuation". *Journal of Mechanisms and Robotics,* ***11***(6).

[2] Vanteddu, T., Sebastian, B., and Ben-Tzvi, P., 2018. "Design optimization of RML glove for improved grasp performance". *ASME 2018 Dynamic Systems and Control Conference, DSCC 2018,* ***1***, pp. 1–8.

[3] Ben-Tzvi, P., Danoff, J., and Ma, Z., 2016. "The design evolution of a sensing and force-feedback exoskeleton robotic glove for hand rehabilitation application". *Journal of Mechanisms and Robotics,* ***8***(5), pp. 1–9.

[4] Chauhan, R., Sebastian, B., Member, S., Ben-tzvi, P., and Member, S., 2019. "Exoskeleton Glove Control". *IEEE Transactions on Human-Machine Systems,* ***PP***(1), pp. 1–10.

[5] Feix, T., Pawlik, R., Schmiedmayer, H.-B., Romero, J., and Kragi, D., 2009. "A comprehensive grasp taxonomy". *Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, pp. 2–3.

[6] Randazzo, L., Iturrate, I., Chavarriaga, R., Leeb, R., and Millan, J. D. R., 2015. "Detecting intention to grasp during reaching movements from EEG". *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS,* ***2015-Novem***, pp. 1115–1118.

[7] Kim, D., Kang, B. B., Kim, K. B., Choi, H., Ha, J., Cho, K.-J., and Jo, S., 2019. "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view". *Science Robotics,* ***4***(26), p. eaav2949.

[8] Wang, X., Tran, P., Callahan, S. M., Wolf, S. L., and Desai, J. P., 2019. "Towards the development of a voice-controlled exoskeleton system for restoring hand function". *2019 International Symposium on Medical Robotics, ISMR 2019,* ***1***, pp. 1–7.

[9] Nagrani, A., Chung, J. S., Xie, W., and Zisserman, A., 2020. "Voxceleb: Large-scale speaker verification in the wild". *Computer Speech and Language,* ***60***, 3, p. 101027.

[10] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., 2017. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications".

[11] Sueur, J., 2014. "A very short introduction to sound analysis for those who like elephant trumpet calls or other wildlife sound". *Muséum national d'Historie naturelle*, pp. 1–17.

[12] Balasingam, M. D., and Kumar, C. S., 2018. "Refining Cosine Distance Features for Robust Speaker Verification". *Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing, ICCSP 2018*(1), pp. 152–155.

[13] Nagraniy, A., Chungy, J. S., and Zisserman, A., 2017. "VoxCeleb: A large-scale speaker identification dataset". *Proceedings of the Annual Conference of the International Speech Communication Association, INTER-*

SPEECH, ***2017-Augus***, pp. 2616–2620.

[14] Liu, W., Wen, Y., Yu, Z., and Yang, M., 2016. "Large-Margin Softmax Loss for Convolutional Neural Networks".

[15] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J., 2018. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* ***40***(6), pp. 1437–1451.

[16] Zhong, Y., Arandjelović, R., and Zisserman, A., 2019. "GhostVLAD for Set-Based Face Recognition". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* ***11362 LNCS***, pp. 35–50.

[17] Chung, J. S., Nagrani, A., and Zisserman, A., 2018. "VoxceleB2: Deep speaker recognition". *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* ***2018-Septe***(i), pp. 1086–1090.

[18] Chellappa, R., Veeraraghavan, A., Ramanathan, N., Yam, C.-Y., Nixon, M. S., Elgammal, A., Boyd, J. E., Little, J. J., Lynnerup, N., Larsen, P. K., and Reynolds, D., 2009. "Gaussian Mixture Models". *Encyclopedia of Biometrics*(2), pp. 659–663.

[19] Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K., 2003. "KNN model-based approach in classification". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* ***2888***, pp. 986–996.

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C., 2018. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520.