Lexa: A Liveness Detection Enabled Voice Assistant

Rodolfo Rodriguez*, Yanyan Li†
Department of Computer Science and Information Systems
California State University San Marcos, San Marcos, California 92096
*rodri833@cougars.csusm.edu, †yali@csusm.edu

***Abstract*—Virtual voice assistants are becoming more prevalent in the modern household or office environment, with the amount of features and account privileges they have rising steadily as well, this along with their lack of traditional authentication methods makes them susceptible to potential attack. This paper proposes a novel approach of creating a liveness detection system for voice activated smart assistants. Smart assistants are targets of replayed audio attacks, a type of computer intrusion that focuses around playing back an either synthetic or pre-recorded audio command to a smart assistant. This approach is novel in its use of spectrogram images to treat this as a image classification problem. We implement transfer learning techniques to change the domain of popular image classification model MobileNetV2, to act as a binary classifier of replayed or genuine voice samples.We performed the final evaluation for this device in a real world test case by implementing the liveness detection system on a raspberry pi based voice assistant. The model created is shown to be successful in protecting a smart assistant from replayed audio attacks to a comparable level to other models created for this task.**

I. Introduction

Industry trends and funding are being directed towards internet of things devices with a central focus in cross-device integration. Often these scenarios are interlaced by a central smart voice assistant. Contemporary manufacturers implement their own systems that interweave these types of devices as a central point, the smart voice assistant very quickly can become a common target for malicious agents.

Modern attacks are already beginning to break headlines. Despite being used with benign intentions, fast food restaurant Burger King has proven that these types of attacks are real. [1] During an advertisement for the fast food chain, a Google assistant command was given asking "Hey Google, What's in a Whopper" and prompting any Google assistant device within earshot of the advertisement to recognize and begin acting upon this command. In response, most household Google Home devices would proceeded to read the Wikipedia article describing the Whopper burger. Google and Wikipedia both took action to prevent this type of abuse going forward, and Burger King made a statement that they would not make this type of advertisement again, but it did act as a moment of realization for many that these devices offer a false sense of security at best and a simple advertisement from a television screen could effortlessly command a presence in the privacy of anyone's home. Teams have also shown that Smart assistants are subject to attacks that are in audible to the human ear [2]. A team from Zhejiang University showed that using ultrasonic frequencies they could consistently send commands while undetected by the end user, this could lead to far more malicious attacks.

These voice assistants are usually configured with rights and privileges to access and manipulate the multiple smart devices in the workplace or home environment. They use user bio-metric authentication features that are centered around the concept of matching voice patterns to the correct user. However, common security authorities are now considering another potential threat, this threat is focused on replayed audio attacks or synthesized audio attacks. This audio could match the voice patterns of the user and offer unauthorized access despite being played back from an audio device. A replayed attack scenario could be an instance where a user speaks, uttering a command to their smart assistant, while a malicious agent discreetly captures this utterance with a hidden audio device. This audio can then be manipulated, or used as is, to utter a similar or separate command when played back to a smart assistant and will often be accepted as a valid command. Countermeasures that have been utilized against this type of threat typically leverage the strengths of deep learning neural networks to determine voice liveness. Models created for this task analyze the audio data captured from these smart voice assistants. Speed and processing limitations must be taken into consideration when designing a system of protection for devices as they are typically low power. The speed of image classification has increased with models being developed for low powered devices as the target platform, such as mobile phones. Our implementation leverages the speed and accuracy of popular image classification models by converting these audio samples into spectrogram images. Upon visual inspection of replayed voice sample spectrograms, noise and visual artifacts can been found frequently, this trend was a inspiring force behind this choice to use spectrogram based classification. Our model attempts to use the image classification base to view this type of noise and use it as a determining factor of authentication. This allows for a low power image classification model to perform in this domain via transfer learning. In this paper, we will be using Mobilenetv2 as our image classifier [3].

Along with the system that performs the classification, our novel approach has a demonstrative example. Through the use of Amazon's AVS SDK, this is supplied as a method for device makers to integrate Amazon Echo features in their products, we have setup an Amazon Echo device to run on

a raspberry Pi running the Raspbian OS. Along with the aforementioned there is also a supplementary software we have dubbed "Lexa" which acts as a liveness detection layer, we have a example of functioning smart assistant with liveness detection protection. The AVS SDK's architectural limitations prevented the system to work with traditional wake word detection, instead opting for a push to talk system. Audio captured by Lexa is converted to a spectrogram image then processed by our machine learning model. At this point a determination is made on whether it is a genuine or replayed sample, with only genuine samples then being fed into the AVS SDK via virtual microphone where it is then processed by Amazon servers for traditional voice command input.

A. Significant Contributions

- Created the first spectrogram based liveness detection system
- Implemented a liveness detection system using transfer learning principles
- Created a real world testing environment with raspberry pi
- Tested the system at with many types of commands at various ranges

II. Background and Related Work

A. Smart Voice Assistant

The use of smart voice assistants such as the Amazon Echo or Google Home are becoming more prevalent in homes across the world. Over 4.2 billion voice assistant devices are in use worldwide, with a projected 8.4 by 2024 [4]. The types of commands that these devices are programmed to receive and recognize are also rapidly expanding within this same market, this can be shown with each devices' own respective catalog for new commands to be created and downloaded by users. These devices can have access to private user data, as well as access to privileged account rights, such as purchasing rights, positional tracking information. Home users who purchase them are looking for a device that can listen to their commands and perform the tasks or answer the questions asked. These devices typically contain a multidimensional microphone for voice capture at long ranges. In addition, they also possess a rolling buffer audio stream that awaits a pre-defined wake word before streaming the audio to a server side recognition system. This is where the commands are translated from audio sample into computer readable commands. For the purpose of this article when discussing smart voice assistants, we are referring the Amazon "Alexa" smart assistant system by the name Echo.

B. Voice Assistant Security Issues

With the current implemented of Echo there is user rights protections. This is done as voice pattern matching that attempts to recognize the user's voice to associate the correct permissions to the commands recognized from these devices. Nevertheless, these protections are limited in scope and don't take into account the potential of a replayed audio attack. Such an attack applies a strategy where the malicious agent captures audio samples of the victim with a recording device. This audio could be spliced with other samples of the victim's voice to create a new command. Voice synthesis is also a potential method to create a sample that matches the voice patterns of a user. The most simple attack would be a straight forward replay of the initial voice command at another time. They then would play back the audio captured to the Echo which would run the commands with the victim's user rights. This would allow the malicious agent to take advantage of privileged rights of the victim on their assistant device.

C. Voice Liveness Detection

Bio-metric authentication such as fingerprint and facial image scanning include security provisions to ensure the user is alive [5], [6]. In the case of voice authentication there are different challenges in the determination that the audio is being uttered by a live person. Methods for determining the liveness of a voice sample have been developed using neural network models [7]–[11]. While nearly all methods of liveness detection use deep learning, they prioritize different aspects of humanity to focus on for their classification. Some use phoneme sounds based on acoustics of the speakers mouth and throat, while others use a series of checks including user movement breathing habits and voice synchronization [12]–[16]. These methods have their limitations with some needing the device to stay in very specific areas near the speaker, and others needing a multitude of data points to be collected. The key distinction in our methodology verses other implementations is the use of a image classifier with the sound files being converted into spectrogram images. All these method of determining liveness has been shown to be effective; yet, two key considerations must be made to ensure the effectiveness and maintain usability in the device. Overall speed of authentication must be rapid, and the accuracy must be high enough to rely on.

D. MobileNet Deep Learning

The MobileNetV2 (MN2) deep learning model has been shown to be effective in use cases where there is limited processing power such as mobile devices. It is a image classification model trained on the ImageNet data-set, that has shown great levels of accuracy. As such if we are able to construct a method where we can take audio samples and process them in such a manner that they could be evaluated as an image, we could leverage the accuracy of the MN2 model in determining whether the audio samples are from a live user or replayed audio attack.

The MN2 architecture is based around the use of what is called Bottleneck residual blocks, these take the place of the standard convolutional layer in an image classifier Fig. 1. The standard convolution is broken into three sections,a 1x1 "expansion layer that will increase the tensor size, followed by a 3x3 depthwise convolution, then finishing with a 1x1 linear "projection" convolution that will decrease the size back down before the next bottleneck layer. This approximates the results

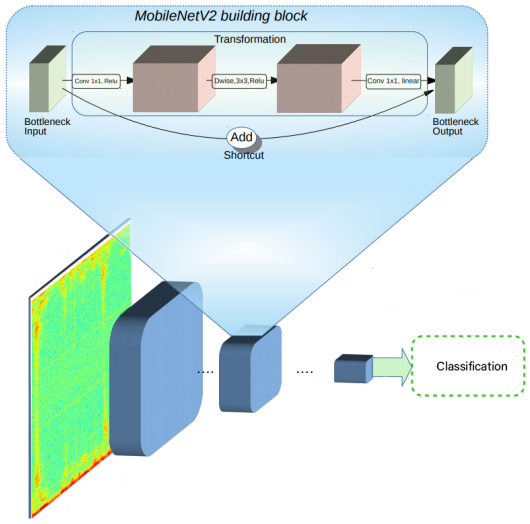
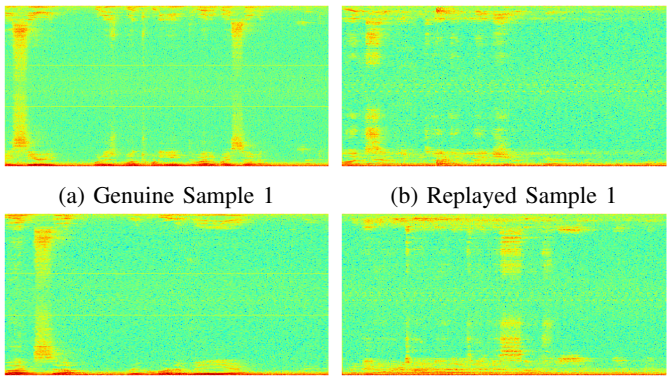


Fig. 1: BottleNeck Residual Blocks, *by Mark Sandler and Andrew Howard, Google Research* [17]

of a standard convolution, but does so significantly faster. After the expansion and depthwise convolutions a ReLU6 activation layer is utilized. Another key aspect of the MN2 architecture is the inclusion of a "depth multiplier" hyper-parameter. This parameter modifies the amount of channels used per layer, which can significantly reduce the amount of computations needed, allowing the model to cut down on training time or run on low power devices.

III. Spectrogram based Liveness Detection

A. Spectrograms



(a) Genuine Sample 1 (b) Replayed Sample 1

(c) Genuine Sample 2 (d) Replayed Sample 2

Fig. 2: Noise can be seen to be added in the replayed sample.

A spectrogram is a 2D graphical view of a sound file overtime, created via Short-Time Fourier Transformation, they maintain data with minimal loss . They acted as a motivating example in attempting to use a image classifier to view audio samples. When viewing spectrogram images of replayed attacks abnormalities become more apparent. Figure 2 contains genuine samples and their corresponding replayed sample. In this example we can see dotted lines run horizontally across both replayed images, likely a result of interference or result of recording frequency. This along with a lack of depth in the middle frequencies can be seen consistently in the replayed samples. There are a key reasons we chose to move forward with spectrogram based classification. The first being the availability of many different image based classification models. many major contributions have been made towards the image classification task in machine learning, as such if we could pivot our problem to fit within those constraints we would be able to capitalize on the accuracy of these models. This would also allow us to create a system that is low power and quick, as most image classification is done on very low resolution images, we can do a lot with limited memory and processing power, this is significant because in this task we aimed to get results within less than a second.

B. ReMASC

The ReMASC is a free to use database from the university of Notre Dame published in September 2019. It is a collection of sample audio files that contain genuine and replayed recording samples. The voice samples include commands for different voice assistant wake words. Four different recording devices were used , all of which are multiple microphone arrays that mimic the types of microphone arrays seen in modern voice assistant devices. This enables the sound data to more closely match that of which can be expected from an off the shelf voice assistant to capture. Their capture environments include two indoor, one outdoor and, a vehicle based environment. The different capture environments assist in giving a wider range of samples while still maintaining realistic expectation of where these devices can be expected used. Finally the replayed audio attacks use a multitude of recording and playback devices, this enables any model trained with this data-set to be flexible when dealing with different playback frequency ranges. When we began the training process for our model, we selected this data set for all the above stated attributes. This would enable our model to be far more robust. We chose to include samples that were labeled as "indoor" as our testing environment was going to exclusively take place within an indoor environment. Before training all samples that would be used we're converted into spectrogram images with a normalized resolution. We not only utilized this data-set for training, it was used for the testing and validation process before we implemented our system in a real world scenario.

C. Transfer learning

The principle idea behind transfer learning is the use a model that has been trained and shown to perform well in one task, then adapt or transfer that model to perform a different task [18]. This can significantly reduce training time and the amount of training samples needs to train a model. In our use case, we implemented the aforementioned mobilenetv2 model, while leveraging the ReMASC data-set for training data on wake word and command utterances [19]. We treated this a binary classification task, labeling samples as either

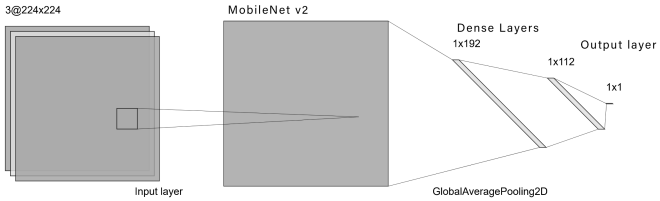


Fig. 3: Lexa Model Architecture

genuine or replayed for training. The audio samples selected were converted to spectrogram images prior to training. The spectrograms images were scaled down to a resolution of 224x224x3 using bicubic interpolation, this is done to match the input requirements of MN2's original training environment Fig. 8. A total of 15062 images were used for training with a ratio of .90 for training and .10 for validation Fig. 6. The MN2 model was used as a base upon which we added dense layers for final classification, the base section was frozen to prevent overwriting of prior training. Hyperparameters were selected from a range of 20 candidate sets. The Random search method was used with a range of parameters being shuffled per iteration [20]. The model used a Binary Cross-entropy for the loss function. The model was trained with a total of 20 epochs, then rolled back to the epoch value with the highest binary-accuracy, this was 8 epochs. This is done as a preventative measure for the over fitting problem [21]. At the output of the MN2 base model an average pooling method is used before outputting to a 192 node dense layer.This output to a 112 dense layer, both using ReLu as the activation function. Final layer is an output node with a sigmoid activation function, this is done so we can gather a probability from the model, being represented as a number between 0-1 Fig. 3. This probability was compared against the determination threshold of .50, with scores above this representing a replayed attack. Evaluation results with an unbiased sample set had a binary accuracy of 85% and Equal Error Rate of 18.9% Fig. 4.

TABLE I: Equal Error Rate Values

	EER Table		
	Lex Real World	***NN-Multichannel***	***Lex Evaluation***
EER	12.5	14.9[a]	18.9

[a] [7]Model also trained with outdoor or vehicle samples.

IV. Lexa Voice Assistant

A result of an effort to create a real world example, we create an implementation of a liveness detection protected echo as proof of concept and testing platform. Due to these type of devices being commercially sold products, they are closed source in most scenarios and difficult to test with. The AVS SDK; however, does offer the Amazon Echo software to be run on non Amazon hardware in an effort to enable third-party manufacturers to create Echo powered devices such as

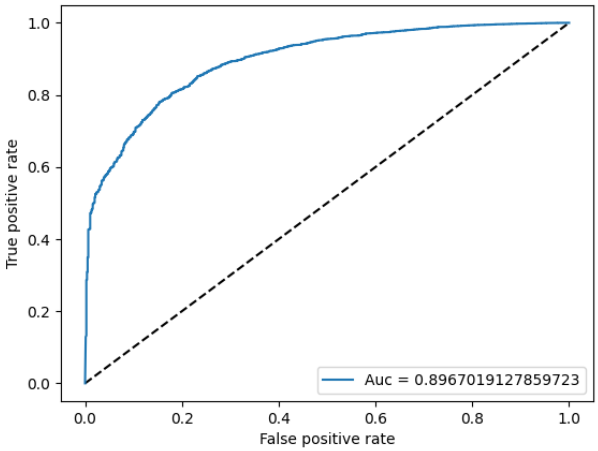


Fig. 4: ROC Curve of Evaluation results

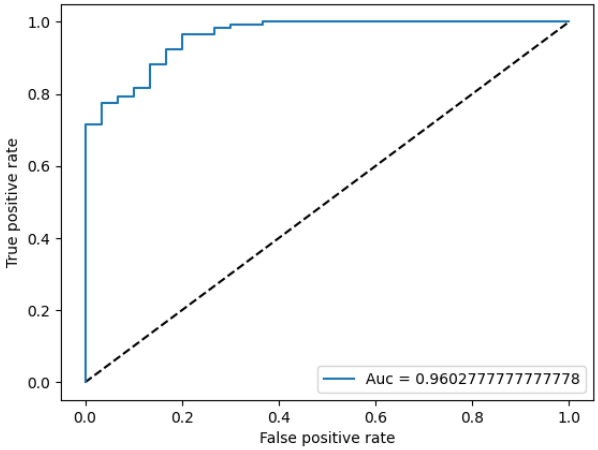


Fig. 5: ROC Curve of experiment results

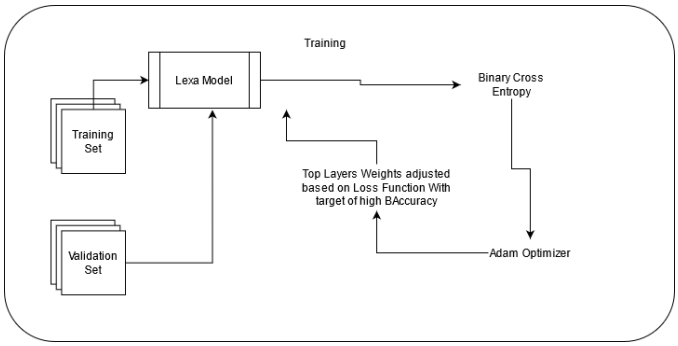


Fig. 6: Training Methodology

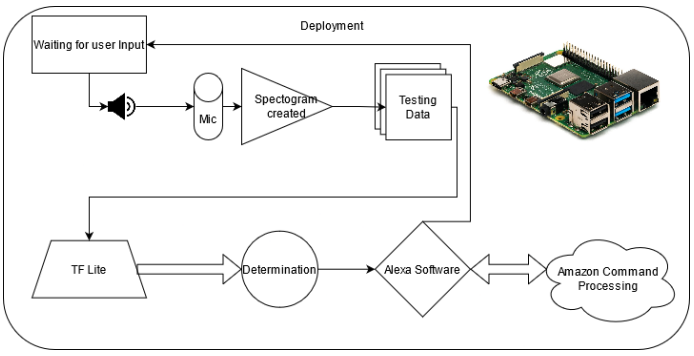


Fig. 7: Deployment Design

smart speakers. We setup this software to on a Raspberry Pi 4, running Raspbian OS version 10(Buster). We added an accompanying software to run in conjunction with this Echo device, the software was dubbed Lexa. It functions as a layer above the Echo, capturing audio commands directed to the Echo and performing the determination on whether it is a genuine or replayed utterance prior command execution. The audio is converted to spectrogram and pre-processed in a similar fashion to that in training in moments after it is captured. From this point a tensorflowLite instance with our transfer learning model loaded takes the spectrogram as input Fig. 7. An output value between 0 and 1 is given. The threshold for determination was decided to be 0.618, this is the threshold where the false acceptance rate and false rejection rate intersect, this can be viewed as the Equal Error Rate(EER) and will be used for comparison to other models. If a determination is below the threshold the audio will be deemed genuine and forwarded to the Echo system so it can begin parsing and acting on the command. Otherwise it will be deemed a replayed attack, a printed message will state that the utterance is a replay.

V. Real World Evaluation

The testing environment included a USB capture microphone that acted as the main interaction point for Echo commands Fig. 10. The capture microphone had a sample rate of 96khz and bit depth of 24. There are also three microphones used as ”attack” microphones, these microphones captured the audio that would be replayed to the capture microphone in an attack. A headset style microphone at 48khz 16 bit depth(A), two channel beam forming webcam style microphone at 32khz 16 bit depth (B), and a smart phone microphone at 48khz 16 bit depth(C). Samples of ten typical smart assistant commands were captured with these attack microphones. Microphones A and C captured a series of samples at a range of less than 4 inches from the subject, these are an effort to capture audio in an unrealistic range to show a best case scenario. A series of samples captured from microphones B and C were captured at a range of 3 ft from the subject, this is an effort to capture audio from the subject at a more realistic range for this type of attack. Two playback devices were used for audio playback to the capture microphone. The first is the playback speakers from the same smartphone used in the capture scenario (D). The second playback are desktop loudspeakers(E) Fig. 9. With 10 utterances per microphone scenario plus an additional 10 for human speaker we total to 50 sample. Samples ranged from 6 to 12 seconds in length. II

A goal of our testing was to determine the effectiveness at different ranges, so the playback devices were placed at three different ranges from the capture microphone. The three ranges were 1 ft, 3 ft , and 5 ft. These ranges are used to re-enact realistic attack scenarios Fig.10.

- The shortest range is a worst case scenario attack, where the attacker is placing the playback device as close as they can.
- the mid-range scenario is similar to the TV advertisement attack mentioned above.
- The longest range could be a attacker who is not within the home but within earshot of the Alexa device from outside.

Samples gathered from microphone C were used for playback from speaker D. Samples gathered from microphone A and B were used for speaker E. The subject also uttered the same commands from the same ranges as a control. A result of of increasing the range was a increase in replayed attack detection accuracy.

TABLE II: Binary Accuracy of Individual test environments

	1 Ft	3 Ft	5 Ft
Genuine Samples	90%	80%	80%
Recording:A Playback:E	100%	100%	100%
Recording:B Playback:E	90%	90%	90%
Recording:C1 Playback:D	80%	90%	100%
Recording:C2 Playback:D	80%	90%	100%

Results from testing showed binary accuracy of 90% Tab II. The EER from real world testing was 12.5% Tab I. This is likely lower than the evaluation accuracy of 18.9% due to limited sample size Fig. 5.

VI. Discussion and Future Work

The current method of forwarding commands to the Amazon smart assistant centers around the use of a virtual microphone. This method is tedious to setup, and introduces a delay equivalent to the length of the spoken command. A method that allows for audio parsing to begin on the Echo while awaiting for liveness detection before execution would reduce this time significantly. This could be accomplished via deeper integration within the AVS architecture, or via direct support from smart assistant manufacturers. The MN2 model base was not trained with the data-set, leaving only the dense layers and output layer to be trained. the MN2 base could be trained to adjust the filters at the lower layers of the architecture this could allow for increased accuracy. The hyper parameters included in the random search were limited

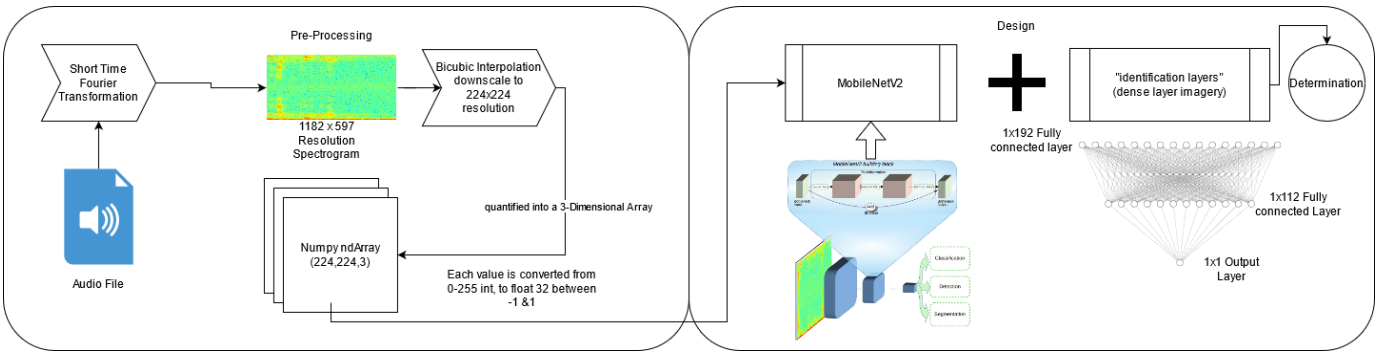


Fig. 8: Pre-processing and Model design

in scope due to limited access to processing time, an increased range of parameters and scope would likely result in a higher level of accuracy.

VII. Conclusion

In this paper we have introduced a novel approach to creating an replay attack detection model that leverages image classification models that have already been shown to excel in identification tasks. This method shows comparable levels of accuracy to other models in the same task. Training time and the amount of samples needed can be decreased significantly. While also creating a proof of concept device that implements this model in a real world scenario.

References

[1] S. Maheshwari, "Burger King 'O.K. Google' Ad Doesn't Seem O.K. With Google," The New York Times, Apr. 12, 2017.

[2] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17, 2017, doi: 10.1145/3133956.3134052.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4520–4520, 2018.

[4] L. S. Vailshery, "Number of voice assistants in use worldwide 2019-2024," *Statista*, January 22, 2021. [Online]. Available: https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/. [Accessed: 07-May-2021].

[5] M. Saini, "Liveness Detection for Face Recognition in Biometrics: A Review," *IOSR Journal of Computer Engineering*, vol. 02, no. 02, pp. 31–36, 2016.

[6] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, "Fingerprint liveness detection based on quality measures," *2009 First IEEE International Conference on Biometrics, Identity and Security* (BIdS), pp. 1–8, 2009.

[7] Y. Gong, J. Yang and C. Poellabauer, "Detecting Replay Attacks Using Multi-Channel Audio: A Neural Network-Based Method," *IEEE Signal Processing Letters*, vol. 27, pp. 920-924, 2020.

[8] M. E. Ahmed, I.-Y. Kwak, J. H. Huh, and I. Kim, "Void: A fast and light voice liveness detection system," *USENIX Security Symposium*, 2020.

[9] L. Zhang, S. Tan, J. Yang, and Y. Chen, "VoiceLive: A Phoneme Localization based Liveness Detection for Voice Authentication on Smartphones," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016

[10] L. Zhang, S. Tan, and J. Yang, "Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017.

[11] J. Shang, S. Chen and J. Wu, "SRVoice: A Robust Sparse Representation-Based Liveness Detection System," 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018, pp. 291-298, doi: 10.1109/PADSW.2018.8644547.

[12] J. Shang, S. Chen and J. Wu, "Defending Against Voice Spoofing: A Robust Software-Based Liveness Detection System," 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2018, pp. 28-36, doi: 10.1109/MASS.2018.00016.

[13] S. Pradhan, W. Sun, G. Baig, and L. Qiu, "Combating Replay Attacks Against Voice Assistants," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 3, pp. 1–26, 2019.

[14] Y. Wang, W. Cai, T. Gu, W. Shao, Y. Li, and Y. Yu, "Secure Your Voice: An Oral Airflow-Based Continuous Liveness Detection for Voice Assistants," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 4, pp. 1–28, 2019.

[15] Y. Lee, Y. Zhao, J. Zeng, K. Lee, N. Zhang, F. H. Shezan, Y. Tian, K. Chen, and X. F. Wang, "Using Sonar for Liveness Detection to Protect Smart Speakers against Remote Attackers," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 4, no. 1, pp. 1–28, 2020.

[16] S. Chen et al., "You Can Hear But You Cannot Steal: Defending Against Voice Impersonation Attacks on Smartphones," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 183-195, doi: 10.1109/ICDCS.2017.133.

[17] M. Networks, "MobileNetV2: The Next Generation of On-Device Computer Vision Networks", Google AI Blog, 2021. [Online]. Available: https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html. [Accessed: 26- May- 2021].

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," Cornell University , 2019.

[19] Y. Gong, J. Yang, J. Huber, M. MacKnight, and C. Poellabauer, "ReMASC: Realistic Replay Attack Corpus for Voice Controlled Systems," *Interspeech 2019*, pp. 2355–2359, 2019.

[20] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn.*, vol. 13, pp. 281–305, 2012.

[21] J. Brownlee, "Use Early Stopping to Halt the Training of Neural Networks At the Right Time," Machine Learning Mastery, 25-Aug-2020. [Online]. Available: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/. [Accessed: 07-May-2021].

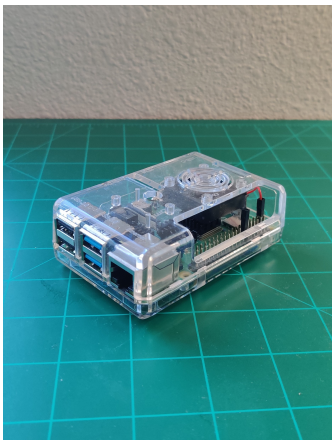
(a) Headset microphone

(b) Beam-forming Webcam Microphone

(c) Smartphone Microphone and Speaker

(d) Lexa Microphone

(e) Loud Speaker



(f) Raspberry Pi

Fig. 9: Devices

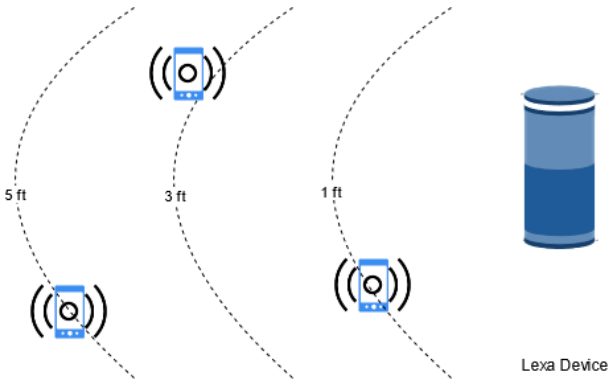


Fig. 10: Testing environment ranges.

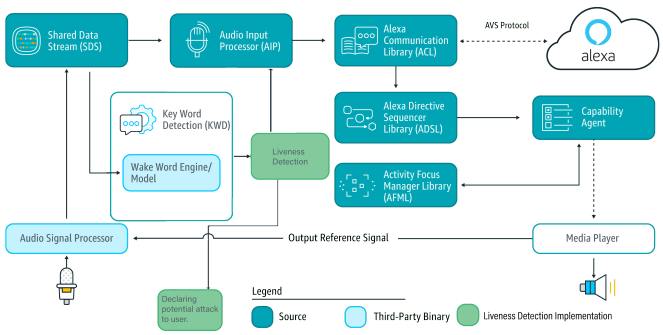


Fig. 11: Modified AVS for future work.