DashCam Pay: A System for In-vehicle Payments Using Face and Voice

Cori Tymoszek*, Sunpreet S. Arora†, Kim Wagner†, and Anil K. Jain*
†Visa Research, Palo Alto CA 94306
*Michigan State University, East Lansing MI 48824

Abstract

We present an open loop system, called DashCam Pay, that enables in-vehicle payments using face and voice biometrics. The system uses a plug-and-play device (dashcam) mounted in the vehicle to capture face images and voice commands of passengers. The dashcam is connected to mobile devices of passengers sitting in the vehicle, and uses privacy-preserving biometric comparison techniques to compare the biometric data captured by the dashcam with the biometric data enrolled on the users' mobile devices to determine the payer. Once the payer is verified, payment is initiated via the mobile device of the payer. For initial feasibility analysis, we collected data from 20 different subjects at two different sites using a commercially available dashcam, and evaluated open-source biometric algorithms on the collected data. Subsequently, we built an android prototype of the proposed system using open-source software packages to demonstrate the utility of the proposed system in facilitating secure in-vehicle payments. DashCam Pay can be integrated either by dashcam or vehicle manufacturers to enable open loop in-vehicle payments. We also discuss the applicability of the system to other payments scenarios, such as in-store payments.

I. Introduction

The payments industry has witnessed the introduction of several innovative payment solutions in the last decade. Examples include contactless payment cards [1], smartphone and smartwatch payment applications (e.g., Apple Pay [2], Google Pay [3], Samsung Pay [4]), and mobile terminals (e.g., Square terminal [5]). While cash and cards remain the two most prominent payment methods, such innovative solutions are being increasingly created and used for conducting in-store and online payments [6]. A relatively new focus area for the payment industry is in-vehicle payments. For in-vehicle payments, the goal is to enable the driver (and/or passengers) to pay in a seamless and secure manner.

A. Existing Solutions

Several vehicle manufacturers have integrated digital assistants, such as Siri and Alexa, in infotainment systems that can be used for making voice-based in-vehicle payments [7]. In fact, Amazon recently piloted in-vehicle payments using Alexa [8]. In addition, some newer vehicles are connected and have integrated payment capabilities in the head-unit [9]. Currently, such systems are closed-loop and typically only enable payments for the owner of the vehicle (and not for passengers in an open loop manner). Also, these systems do not enable owners of older vehicles that lack these advanced capabilities to conduct in-vehicle payments. Alternatively, smartphone apps can be used for making in-vehicle payments. While they are a viable option for passengers, smartphone usage can be quite distracting and is not recommended for drivers.

B. Proposed Solution

Given the limitations of existing payment solutions, we design an open loop system that uses face and voice biometrics for enabling in-vehicle payments (Fig. 2). The users of the system enroll their face and voice templates once in a mobile app, and then pair their mobile device with a plug-and-play device (i.e., a dashcam) that is
mounted in the vehicle. To initiate payment, a passenger invokes the dashcam with a trigger phrase, for example, Hey DashCam and then issues a command, such as Pay for gas at pump 5. On hearing the trigger phrase, the dashcam takes a picture of the passengers sitting in the vehicle using the in-cabin camera, as well as an audio recording of the command. The dashcam then initiates a privacy-preserving biometric comparison protocol with each of the connected mobile devices of the driver and/or passengers. Both face and voice biometrics are used to determine which passenger issued the command and wants to pay. If a unique payer is determined, payment is initiated via the mobile device of the payer.

We collected data from 20 different subjects at two different sites using a commercially available dashcam (Fig. [1]), and evaluated open-source biometric recognition algorithms to show the feasibility of our system. We also developed an Android-based prototype of our system using open-source software packages to show the utility of DashCam Pay for facilitating in-vehicle payments.

C. Payment Use Cases

The proposed system can be useful in a variety of payment scenarios (Fig. [3]). Below, we discuss the benefits of our system for key use cases.

1) Drive-throughs: Cash or cards are still heavily used at drive-throughs for making payments. The typical process can be quite inconvenient given that the driver has to reach for their wallet to retrieve cash or a payment card, and hand this payment to a merchant representative. Some merchants, such as Starbucks, are using payment models where users pay via app [10]. However, these apps may still require the customer to present their mobile device to an employee. In our system, the customer simply uses a voice command and this additional step is eliminated.

2) Toll booths: Road tolls are assessed by stopping at a booth, by registration of a radio-transmitting device placed in the vehicle, or by mail after a camera at the toll road takes a photo of the vehicle’s license plate. With DashCam Pay, the driver could make a payment while approaching the toll booth using a voice command. The license plate number is registered in the dashcam during initial installation. Then, the vehicle’s location is used to determine to which authority the payment should be sent.

3) Parking: A variety of modern parking solutions exist, including multi-space meters, pay-by-phone, pay-and-display, and others [11]. However, all of these require the customer to interact with either a physical meter machine or their mobile device. With DashCam Pay, the customer could simply give a voice command with either their license plate number or space number, as required by the location. This identifier is then sent along with payment to the relevant parking authority (as identified by the device’s location).

4) Gas stations: Our solution allows the customer to skip the step of inserting payment card at a gas station. Instead, the customer can give a voice command that includes the gas station number. The merchant is again determined by the device’s location.

5) Retail stores: Many options exist for making payments at a point-of-sale terminal in a retail store, including traditional swipe or chip cards, contactless cards, and near-field communication (NFC) payments from mobile devices. Our system could be deployed in a similar spirit at payment terminals in retail locations. These terminals would be equipped with a camera and microphone, and would allow customers to make in-store payments without interacting with their mobile device or payment card.

D. Key Contributions

The major contributions of this work are:

- A multi-biometric system based on face and voice that enables secure and seamless in-vehicle payments.
Fig. 3: Illustrations of potential use cases of the DashCam Pay system. Advantages of using the proposed system are listed below each use case.

- An open-loop realization of this system that is not tied to any vehicle manufacturer or to any proprietary payment application; the fundamental system design is reusable for other payment scenarios such as in-store payments.
- A privacy-preserving biometric authentication protocol that compares encrypted biometric data locally between a vehicle-mounted device and nearby mobile devices, driven by a dynamic biometric template gallery based on passengers currently sitting in the vehicle.

II. Payer Identification

Our three key objectives were to build a system that is (1) seamless, (2) secure and (3) responds in real-time. With these in mind, the system design includes the following modules:

- Trigger phrase detection: allowing the user to invoke the dashcam in a hands-free manner by speaking a specific phrase (e.g., "Hey DashCam...")
- Face recognition, allowing (a) enrollment of a user’s face on the user’s personal mobile device, and (b) real-time face recognition on the dashcam device to identify the potential payers from the passengers with mobile devices connected to the dashcam (whether in advance at connection time, periodically throughout the ride, or at the time a voice command is issued)
- Speech recognition: for the parsing and execution of the user’s voice commands

### Feasibility Study

To assess the feasibility of performing biometric recognition in real-life, in-vehicle scenario, we collected data using a commercial dashcam device (Vantrue N2 Pro Uber Dual Dash Cam).

1) Data: The data consists of audio and video streams from 20 subjects in five different vehicles at two different times.

### Table I: Data collected for feasibility experiments.

| #Subjects | #Vehicles | Visible rec. | Infrared rec. | Audio rec. |
|-----------|-----------|--------------|---------------|------------|
| 20        | 5         | 4 mins.      | 2 mins.       | 2 mins.    |

- Speaker recognition: allowing (a) enrollment of the user’s voice on the user’s personal mobile device, and (b) real-time speaker recognition on the dashcam device to determine which of the identified potential payers (using face recognition) is the payer.
- Wireless communication interface: facilitating communication between the users’ personal devices and the dashcam device (such as BLE or Wi-Fi Direct).
- Payment network interface: connecting the users’ devices to the payment network for the release of payment funds to the merchant.
- Cryptography module: for generating and managing homomorphic encryption keys to enable privacy-preserving biometric comparison.

1 Uses Sony Exmor IMX323 sensor with four infrared LEDs for dual 1920x1080p visible and infrared spectrum video.
| Experiment                  | Software         | Performance                  |
|-----------------------------|------------------|------------------------------|
| Face Detection              | MTCNN [12]       | TPR=99.1%, FPR=0.01%         |
| Face Recognition (1:N)      | FaceNet [13]     | TPR=98.9%, FPR=0.01%         |
| Trigger Phrase Detection    | Mycroft AI Precise [14] | TPR=98.2%, FPR=1%         |
| Speech Recognition          | DeepSpeech [15]  | WER=3.65%                   |
| Speaker Recognition (1:N)   | COTS             | TPR=98.4%, FPR=0.01%         |

TABLE II: Feasibility experiment results. TPR = True Positive Rate, FPR = False Positive Rate, WER = Word Error Rate (post dictionary-based correction). N is the number of subjects sitting in the vehicle.

The dashcam’s default settings were used, whereby it automatically switches to infrared wavelength video recording when low lighting levels are detected. For each subject, approximately four minutes of visible spectrum and two minutes of infrared spectrum video were captured. Audio data was captured simultaneously, with scripted verbal commands comprising approximately two minutes of total audio per subject. Each subject was recorded while giving a number of voice commands for four different use cases: fuel, toll, parking and fast food. These commands were constructed specifically for ease of use, and to ensure that the user includes the necessary information for each proposed use case (e.g., parking space number, order number) Example sentences include:

- "Hey DashCam, pay for parking at space number 5208."
- "Hey DashCam, pay for order number 120."
- "Hey DashCam, pay for toll."
- "Hey DashCam, pay for gas at pump six."

2) Evaluation: Next, we evaluated one state-of-the-art biometric recognition algorithm each for face detection, recognition, trigger phrase detection, speech and speaker recognition to determine the feasibility of building the proposed system.

- Face Detection: Pre-trained multi-task convolutional neural network (MTCNN)-based face detector [12] was used for face detection experiments. Face locations of subjects sitting in the vehicle were manually labelled with a bounding box in each frame of a recorded video. Face detection performance was computed in aggregate over all frames in a video. For example, if a face was detected in a location that does not overlap with a manually labelled bounding box it was recorded as 1 false accept. On the other hand, if no face was detected within a bounding box it was recorded as 1 false reject. In our experiments, MTCNN-based method resulted in 0.01% false accepts and 0.9% false rejects on the collected data. Face detection failure was observed in captured frames with (i) extreme facial pose of subjects, or (ii) lack of proper illumination. Figs. 4a and 4b show failure examples. Note, however, that extreme facial pose is less likely to be encountered in practice at the time of recognition as the subjects cooperate while interacting with the dashcam.

- Face Recognition: Pre-trained FaceNet model [13] was used for face recognition experiments. An image of each subject sitting in the vehicle captured using their smartphone along with their subject identifier was assumed to be enrolled in the gallery. Detected faces in each video frame were assumed to be probe images. Subject identifiers for probe images were manually labelled. Small-scale identification experiments (1: N where N is the number of subjects sitting in the vehicle) were performed to assess performance on the collected data. Identification performance was measured using true positive identification rate (TPIR) at a fixed false positive identification rate (FPIR).

In our experiments, FaceNet yielded TPIR of 98.9% at FPIR of 0.01%. Fig. 4b shows failure of face recognition in low lighting environment in the visible spectrum.

- Trigger Phrase Detection: Mycroft AI precise [14] was used for trigger phrase detection experiments. Positive examples corresponding to speech samples containing the trigger phrase, and negative examples referring to speech samples that do not contain the trigger phrase were manually created from the data. Pre-trained models are available for certain trigger phrases. However, to train a custom detector for the phrase "Hey DashCam", audio data was split into training and testing (50-50 split: 10 subjects for training and 10 for testing). For each positive sample, five negative samples were used in the training set to reduce false positives. The trained detector yielded false negative and false positive detection rates of 0.8% and 1%, respectively, on the test data.

- Speech Recognition: Mozilla’s open-source Deep-Speech implementation [15] was used for speech recognition experiments. Audio data extracted from the video streams was manually labelled based on the audio commands spoken by the subjects. Word error rate (WER) which measures incorrectly detected words in the speech was used as the evaluation metric for testing recognition performance. The pre-trained model was first evaluated on the extracted
audio data. Although the pre-trained model achieved 7.5\% WER on the LibriSpeech test-clean benchmark [16], 25.4\% WER was obtained in our evaluation on the collected data. The model frequently failed to recognize relevant words like “dashcam”, “gas”, and “parking”. Because the model performance was below par, audio data was used for fine-tuning the pre-trained model. Cross-validation experiments (20 percent for fine-tuning, and 80 percent for evaluation) were conducted to measure performance after fine-tuning. Post fine-tuning, average WER reduced to 8.7\%. The majority of errors (68\%) were partial; for instance, “parking” recognized as “parting”. Such errors were corrected using a dictionary of permitted words/phrases in a command. The following heuristic was used: if edit distance between the detected word and the closest word in the dictionary was less than or equal to 2, the detected word was auto-corrected to the closest word in the dictionary. Using this approach, overall WER reduced to 3.65\%.

- **Speaker Recognition**: A commercial off-the-shelf system (COTS) was used for speaker recognition experiments. Three samples of an audio command from each subject were enrolled in the system. Small-scale identification experiments (1: N where N is the number of subjects sitting in the vehicle) were performed to assess speaker recognition performance. Identification performance was measured using true positive identification rate (TPIR) at a fixed false positive identification rate (FPIR). TPIR of 98.4\% was obtained at FPIR of 0.01\%.

Table II summarizes the evaluation results. Overall results show the feasibility of conducting in-vehicle payer identification using face and voice biometrics.

### III. Prototype Development

Commercially available dashcams are not programmable, so we decided to implement the prototype as a software stack in Android, wherein one Android device is mounted in-cabin to act as the dashcam, and the others act as personal devices of the passengers (Fig. 5). The software packages that were considered for use are summarized in Table III.

#### A. Wireless networking interface

One basic requirement for the system to work in real-time is a framework to connect the dashcam to any nearby mobile devices that have the payment app installed and configured. This connection needs to be both secure and seamless, occurring with no intervention from the user.

Given this requirement, we selected Google’s Nearby Connections API [25]. This API uses both Bluetooth and Wi-Fi Direct to connect devices. The combination of modalities allows for the high speeds and versatility of Wi-Fi, with Bluetooth as a fallback. These technologies are also platform-agnostic, and can be implemented on any realization of the dashcam device, whether it be a mobile device, or a Raspberry Pi-like device (via Google Things) [26].
Table III: A summary of software packages considered for use in the prototype. Software names in bold indicate those that were selected for use. Note that some limitations, such as licensing fees, are specific to prototype integration.

| Software             | Function                          | Benefits                                                        | Limitations                                                                 |
|----------------------|-----------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------------------|
| Kaldi [17]           | Trigger phrase detection, speech recognition, speaker recognition | Multifunctional, high accuracy [18]                           | No Android library available                                               |
| Mycroft AI Precise [14] | Trigger phrase detection          | Reported to have good performance; good performance in evaluation | No Android library available                                               |
| Kitt.AI Snowboy [20] | Trigger phrase detection          | Reported to have good performance                               | Requires licensing for custom phrase model                                 |
| Picovoice Porcupine [22] | Trigger phrase detection          | Reported to have good performance                               | Requires licensing for custom phrase model                                 |
| FaceNet [13]         | Face recognition                  | Fast and accurate face detection and identification [13] in evaluation, Android implementations available | Pre-trained models for mobile devices are too large for distribution (approx. 200MB) |
| Mozilla Project DeepSpeech | Speech recognition               | High accuracy [18], pre-trained models can be easily fine-tuned, Android implementations supported | Cannot use fine-tuned models on Android (at time of prototype development) |
| Android SpeechRecognizer [23] | Speech recognition               | High accuracy [23], model files are downloaded as part of Android OS | Not available for iOS user devices or non-Android commercial dashcams       |
| Alize [24]           | Speaker recognition               | Android implementation supported                                 | Poor performance in evaluation                                             |
| COTS                 | Speaker recognition               | Good performance in evaluation                                  | Offline implementation not available                                        |

B. Trigger phrase detection

The prototype implementation platform (Android) did not provide functionality for a third-party application (e.g., Mycroft AI precise [14]) to continuously access the device microphone. Therefore, we implemented trigger phrase detection using Google Assistant as a bridge. This allows the user to give a command such as “Hey Google, open DashCam Pay” or even “Hey Google, tell DashCam Pay to pay for gas at pump number four,” which would then trigger the same authentication and payment flow described above.

Software packages that could be useful for platforms other than Android are listed in Table III.

C. Face recognition

FaceNet [13] provided reasonable face recognition performance in our evaluation. Although the available model files for FaceNet are somewhat large in size (approx. 200MB), the model can be ported to the Android platform. Hence, it was used in our prototype. Note that other state-of-the-art models like SphereFace [27] and CosFace [28] can also be used.

D. Speech recognition

The pre-trained DeepSpeech model [15] did not provide adequate performance during our evaluation, so we had to fine-tune the model on the collected data. At the time of prototype development, however, DeepSpeech’s Android library did not support the use of custom models. Therefore, Kaldi and Android’s SpeechRecognizer class [29] were considered as potential alternatives. However, no pre-built Android library was provided for Kaldi, and running the provided binaries on Android required root access. Given this, we chose Android’s SpeechRecognizer class for the prototype, as it did not require additional software, and has the ability to run offline in real-time with state-of-the-art results [23]. Note, however, that this library is Android-specific.

E. Speaker recognition

Open-source SDKs like Kaldi and Alize [24] were initially considered for speaker recognition. As mentioned earlier, for Kaldi, there is no pre-made Android library, and running the binaries on an Android device requires root access. In the case of Alize, an Android implementation exists, but the performance was unsatisfactory during the initial live testing of the Alize demo application for android in a speaker verification scenario. Significant overlap was observed in the scores returned for the genuine and imposter verification attempts.

The initial system design included a fully offline authentication process. However, due to the above challenges with prototype platform compatibility and performance, we implemented the prototypical speaker recognition using a commercial off-the-shelf (COTS) speaker recognition API to demonstrate the system’s feasibility.
E. Privacy-preserving biometric comparison

Our system creates, stores, and compares each user’s biometric templates on local devices (namely, the user’s mobile device and the dashcam), without transmitting the templates to a remote server. Additionally, templates are stored and compared in the encrypted domain. Below we describe the different steps in the privacy preserving biometric comparison protocol used in our system (Fig. 6).

1) Enrollment: At the time of initial enrollment, a user captures their face and voice data on their personal mobile device. Specifically, the user is prompted to take one or more selfie images, and to record voice commands pertaining to different use cases, such as drive-through, parking and gas. Face and voice templates are generated from this data. Additionally, the user device generates a public-private key pair using homomorphic encryption, e.g., Pallier encryption [30], encrypts the face and voice templates using the public key, and stores the encrypted templates. Once enrolled, the user’s device becomes available for connecting to a dashcam over the wireless networking interface. Additionally, the user also explicitly permits pairing their device to a dashcam device.

2) Device discovery: While operating, a dashcam periodically searches for enrolled devices in proximity, and initiates a secure connection with each nearby enrolled device over the wireless networking interface. If the connection is successful, each device sends their encrypted templates and public key to the dashcam. This results in the creation of a dynamic template gallery on the dashcam.

3) Authentication: When a voice command is given, the dashcam captures an audio recording of the command, as well as an image of the vehicle’s cabin. A voice template is generated from the command audio, and a face template is created from each detected face. The authentication is then performed in two steps. First, the templates generated from the detected faces are used to identify candidate payers from the passengers with mobile devices connected to the dashcam. Second, the voice template from the command is used to disambiguate between the candidate payers to determine who issued the command.

Comparisons are performed in the encrypted domain using homomorphic encryption [31] such that the encryption does not impact biometric matching performance, and encrypted similarity scores are generated. Alternatively, secure multi-party computation techniques [32] can be used. Each encrypted similarity score is sent to the corresponding user’s device, where it is decrypted with the user’s private key. The user’s device compares the score to a predetermined threshold, and reports whether a match was found to the dashcam using a zero-knowledge proof [33]. Use of zero knowledge proof results in the verification of the match by the dashcam without exposing the user’s biometric templates or private key.

4) Payment: Once the dashcam verifies that a match is obtained, and, if exactly one matching payer is found, the dashcam instructs the payer’s mobile device to initiate the transaction. If no matching payer is found or more than one matching payer is found, a remedial action is taken. One remedial action when authentication fails, for instance, is to request the user to retry authentication. In the unlikely scenario that authentication fails after multiple attempts, another remedial action is to prompt the user to use an alternative method, or to call a particular number to finish the payment transaction.

IV. Discussion

In addition to the in-vehicle scenario, the proposed system can also be useful for other types of payment. These include both traditional payment scenarios (such as physical stores, as discussed in Section I-C5, and ATMs), as well as “situated payments”. Situated payments are those scenarios where a traditional point-of-sale (POS) terminal or cash register would not typically be installed, but payments may still be needed (e.g., a farmer’s market, vending machine, or, indeed, a vehicle). In the third quarter
of 2019, Square reported a 25% year-over-year increase in gross payment volume, demonstrating that merchants are increasingly using these modern alternatives instead of a traditional cash register. This evolution in POS systems has recently been the focus of industry research as well [34]. Below, we highlight some potential use cases.

- **Hotels and homeshares**: The system could be installed on a kiosk to pay for room service food, pay-per-view television media, snacks, etc.

- **Vending machines**: Existing systems, such as Stockwell cabinets, eliminate physical and tap-based payments for vending machine transactions. In these systems, users register a payment card in the vendor’s mobile application. The user presses a button in the mobile application to unlock the cabinet’s door, and takes the desired products. The machine detects which products have been taken, and debits the card of the user who unlocked the door. Our system could improve upon this by eliminating the mobile application interaction - the user could simply approach the door and speak a command to verify their identity and access the stocked products.

- **ATMs**: The system could be implemented at an ATM to supplement or replace a traditional card and PIN. The user’s device could be used in place of a debit card to initially claim an identity by connecting to the ATM using our framework. Then, the user’s face and voice can be used to validate their identity before providing funds.

- **Self-driving taxis**: As autonomous vehicle technology advances, we may someday be able to hail a self-driving taxi. When entering the vehicle, the passenger may not be known to the taxi company. However, with DashCam Pay installed in the taxi, the passenger’s device can connect to the dashcam to claim the passenger’s identity. Then, the passenger can quickly use their face and voice to verify themselves and provide payment. A passenger’s identity could also be associated with “favorite” destinations, such as their home or workplace, for convenient transportation.

**V. Conclusions and Future Work**

We propose a system for conducting in-vehicle payments using face and voice biometrics. One particularly important feature of the proposed system is its open-loop nature. While proprietary in-vehicle systems can deliver payment capabilities, the proposed system streamlines the user experience. Any merchant or user can participate, without any dependency on payment technology included by a vehicle manufacturer. This alleviates the need for vehicle manufacturers to develop proprietary payment systems.

In the prototype, an Android smartphone was used to mimic a dashcam device. However, in practice, it would be preferable to install this system on a dedicated dashcam device. We invite the research community to integrate the system we present here in an open-source hardware device. Ideally, such a device should be designed with relatively inexpensive, off-the-shelf parts to enable widespread, low-cost deployment.

While open source modules show the feasibility of realizing the proposed system, sourcing commercial software packages for face, speech and speaker recognition is recommended for a production system. This will ensure reasonable system performance and user experience in practice. In addition, the DashCam Pay mobile application needs to interface with merchants in real-time in order to process payments. Furthermore, comprehensive end-to-end testing and large scale real-world pilots are required to refine the system before deployment in production.
Acknowledgements

We would like to thank Lacey Bartlett and Saman Sarraf for their contributions in the initial development phases of this project. We would also like to thank Susan Oppelt for the illustrations shown in Figs. 2 and 5.

References

[1] (2019) FFIEC - contactless payment cards, proximity payments and other devices. Federal Financial Institutions Examination Council. [Online]. Available: https://www.ffiec.gov/dr-handbook/retail-payment-systems/payment-instruments-clearing-settlement/emerging-retail-payment-technologies/contactless-payment-cards/proximity-payments-and-other-devices.aspx

[2] (2019) Apple pay. Apple. [Online]. Available: https://www.apple.com/apple-pay/

[3] (2019) Google pay. Google. [Online]. Available: https://pay.google.com/

[4] (2019) Samsung pay. Samsung. [Online]. Available: https://www.samsung.com/us/samsung-pay/

[5] (2019) Square terminal. Square. [Online]. Available: https://squareup.com/us/en/hardware/terminal

[6] K. Wagner, J. F. Sheets, M. Nelsen, and J. Jin, “Multi-device transaction verification,” Feb. 19, 2019, U.S. Patent App. 10/210,521.

[7] (2019) State of in-car voice assistants. Rain. [Online]. Available: https://rain.agency/state-of-in-car-voice-assistants/

[8] Amazon. (2020) Amazon announces new automotive products and solutions at CES 2020. [Online]. Available: https://blog.aboutamazon.com/devices/new-automotive-products-and-solutions-at-ces-2020

[9] (2019) Shell and general motors deliver nationwide in-dash fuel payment. Shell. [Online]. Available: https://www.shell-us.media/2019-media-releases/shell-and-general-motors-deliver-nationwide-in-dash-fuel-payment.html

[10] (2019) Starbucks app for iphone and android. Starbucks. [Online]. Available: https://www.starbucks.com/coffeehouse/mobile-apps

[11] (2019) Parking meter. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Parking_meter

[12] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7925882

[14] (2019) Mycroft AI. Mycroft AI. [Online]. Available: https://mycroft.ai/

[15] (2019) Project DeepSpeech. Mozilla. [Online]. Available: https://github.com/mozilla/DeepSpeech

[16] R. Morais. (2019) DeepSpeech 0.6: Mozillas speech-to-text engine gets fast, lean, and ubiquitous. [Online]. Available: https://hacks.mozilla.org/2019/12/deepspeech-0-6-mozillas-speech-to-text-engine/

[17] (2019) Kaldi ASR. Kaldi. [Online]. Available: https://kaldi-asr.org/

[18] G. Synnaeve. (2019) WER are we. [Online]. Available: https://github.com/sjhy/wer_are_we

[19] (2019) Software comparison. Mycroft AI. [Online]. Available: https://github.com/MycroftAI/mycroft-precise/wiki/Software-Comparison

[20] (2019) Snowboy. Kitt.AI. [Online]. Available: https://snowboy.kitt.ai/

[21] (2019) Wake word benchmark. Picovoice. [Online]. Available: https://github.com/Picovoice/wake-word-benchmark

[22] (2019) Porcupine. Picovoice. [Online]. Available: https://picovoice.ai/products/porcupine.html

[23] Y. He, T. N. Salnath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, and et al., “Streaming end-to-end speech recognition for mobile devices,” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2019. [Online]. Available: http://dx.doi.org/10.1109/CASSP.2019.8682336

[24] (2019) Alize open source speaker recognition. Laboratoire Informatique d’Avignon. [Online]. Available: https://alize.univ-avignon.fr/

[25] (2019) Nearby Connections API. Google Developers. [Online]. Available: https://developers.google.com/nearby/connections

[26] (2019) Alizé open source speaker recognition. Laboratoire Informatique d’Avignon. [Online]. Available: https://alize.univ-avignon.fr/

[27] D. Antonioli, N. O. Tippenhauer, and K. B. Rasmussen, “Nearby threats: Reversing, analyzing, and attacking Google’s ‘Nearby Connections’ on Android,” in NDSS, 2019.

[28] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” 2017.

[29] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large margin cosine loss for deep face recognition,” 2018.

[30] (2019) Speech recognizer. Android Developers. [Online]. Available: https://developer.android.com/reference/android/speech/SpeechRecognizer

[31] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Advances in Cryptology — EUROCRYPT ’99, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238.

[32] V. Naresh Boddeti, “Secure face matching using fully homomorphic encryption,” 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), Oct 2018. [Online]. Available: http://dx.doi.org/10.1109/BTAS.2018.8698601

[33] J. Blackhurst, K. Wagner, J. Sheets, S. S. Arora, and C. Jiang, “Use of biometrics and privacy preserving methods to authenticate account holders online,” U.S. Patent WO2 2019 032 301A1, Feb. 02, 2019.

[34] S. Goldwasser, S. Micalli, and C. Rackoff, “The knowledge complexity of interactive proof-systems,” in Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, ser. STOC 85. New York, NY, USA: Association for Computing Machinery, 1985, p. 291304. [Online]. Available: https://doi.org/10.1145/221415.22178

[35] E. Starr, D. F. Olson, A. Dawson, A. Jimenez, J. He, A. Sisodiya, J. A. T. P. Palacios, B. P. Parikh, S. V. Mahajan, S. Vuddaraju, S. S. Tadepalli, L. Best-Rowden, K. R. Wagner, S. S. Arora, and S. Lohtia, “Authentication based on biometric identification parameter of an individual for payment transaction,” U.S. Patent Application US20 190 220 866A1, Jun. 13, 2019.