Zewoudie, Abraham; Bäckström, Tom

Federated Learning for Privacy Preserving On-Device Speaker Recognition

Published in:
ISCA Symposium on Security and Privacy in Speech Communication proceedings

DOI:
10.21437/SPSC.2021-1

Published: 01/01/2021

Document Version
Publisher's PDF, also known as Version of record

2021 ISCA Symposium on Security and Privacy in Speech Communication
10-12 November 2021, Online

Federated Learning for Privacy Preserving On-Device Speaker Recognition

Abraham Woubie and Tom Bäckström

Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland

abraham.zewoudie@aalto.fi, tom.backstrom@aalto.fi

Abstract

State-of-the-art speaker recognition systems are usually trained on a single computer using speech data collected from multiple users. However, these speech samples may contain private information which users are not willing to share. To overcome such potential breaches of privacy, we investigate the use of federated learning in speaker recognition. Distributed learning methods such as federated learning enable us to train a shared model without sharing the private data by training the models on edge devices where the data resides. In the proposed system, each edge device trains an individual model which is subsequently sent to a secure aggregator. To provide contrasting data without the need for transmitting data, we use a generative adversarial network (GAN) to generate impostor data at the edge. Afterwards, the secure aggregator merges the individual models, builds a global model and transmits the global model to the edge devices through a main server. Experimental results on the Voxceleb-1 dataset show that the use of federated learning for speaker recognition system provides two advantages. Firstly, it retains privacy since the raw data does not leave the edge devices. Secondly, experimental results show that the aggregated model provides better average equal error rate than the individual models.

Index Terms: edge computation, federate learning, privacy, secure aggregator, speaker recognition

1. Introduction

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual identifying information included in speech signals. It is divided into speaker identification and speaker verification; Speaker identification determines which registered speaker provides a given utterance from amongst a set of known speakers [1]. Speaker verification accepts or rejects the identity claim of a speaker [2]. Thus, speaker recognition is a vital feature when providing access to private services, as actions should be triggered only when a user with sufficient access-rights has been identified.

Machine learning methods have recently been successfully used in a wide range of applications including speaker recognition [3]. Machine learning techniques usually train a deep network model by using a large number of labeled training data samples. The data samples are often collected on end-devices such as smartphones and the model is trained using a computationally powerful centralized server [4]. The users send their data to the server which in turn uses the huge amount of data it collected from different uses to train a generic deep neural network model. However, the users' data may contain private information which the users prefer not to share.

The transmission of large amounts of training data from each user to the server may also bring a substantial load on the communication link. This creates the need to train the model on the individual devices, i.e., train a centralized model in a distributed fashion [5]. The federated learning technique proposed in [6] can be used to update such decentralized models. Federated learning is a distributed machine learning method where a model can be trained on a large corpus of decentralized data. Thus, instead of requiring the users to share their private data, each user trains the network locally, and sends only updates to its locally trained model to the server. Afterwards, the server aggregates these updates into a global model [7, 8], commonly using a weighted average, known as federated averaging [6].

Privacy concerns are considered as one of the major challenges in speaker recognition application [9] as it involves the complete sharing of speech data, which can bring threatening consequences to people's privacy. Federated learning is a promising technique to avoid privacy infringement by involving multiple participants to collaboratively learn a shared model without revealing their local data. For example, many commercial and public organizations want to take utmost care to uphold user privacy. This makes federated learning an important approach to consider when dealing with data that is private since it protects users' privacy.

The main contribution of this work is the application of federated learning in the training of a deep neural network based speaker recognition classifier with the objective of preserving user privacy. In the proposed system, each device trains its own local model locally, and sends updates to the local model to a secure aggregator. The secure aggregator aggregates the local models from the different devices, builds a global model and sends it to the main server. Finally, the main server sends the global model to each device.

The proposed system enables training of a speaker recognition model based on a deep neural network, using data stored on the devices which will never leave the corresponding device. The models are combined in the cloud with federated averaging, constructing a global model which is pushed back to devices for inference. The implementation of secure aggregation ensures that on a global level individual updates from devices are uninspectable. Since edge devices transmit only model-updates, raw data never leaves the edge. The aggregator thus has access only to a model trained to identify a local speaker and all other information about speech signals remains at the edge.

A second novelty is the use of a generative adversarial network (GAN) to generate impostor data at edge devices. By using a GAN, we avoid the need to transmit impostor data to the edge or collect such data at the edge. Transmission of impostor data could be a significant burden on bandwidth, but, importantly, could also provide access to differential information about the local speaker. Collection of impostor data at the edge could, on the other hand, be impractical.

Our experimental results on Voxceleb-1 dataset show that speaker recognition could benefit from federated learning by not exporting sensitive user data to central servers, while achieving promising results compared to individual local models.

The rest of this paper is organized as follows. Section 2 describes the architecture of the proposed system. Experimental

10.21437/SPSC.2021-1

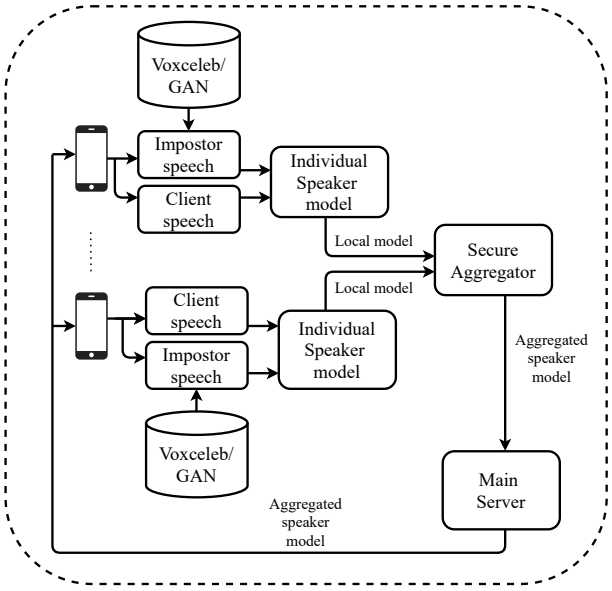

Figure 1: *The proposed speaker recognition system using federated learning. The use of the secure aggregator is that a group of mutually distrustful devices having private data collaborate to compute an aggregate value without revealing their private data.*

results and conclusions are presented in Section 3 and Section 4, respectively.

2. Proposed System

Federated learning enables distributed training of speaker recognition models with heterogeneous clients. As it is shown in Fig.1, the proposed federated learning for speaker recognition has three main components: 1. Edge devices, 2. a secure aggregator and 3. a main server. The edge devices can be for example mobile phones, laptops, or similar devices, while the aggregator and main server are typically cloud services.

Figure 1 shows that a central model is trained over a distributed dataset, where a large number of nodes (e.g. user devices) hold variable-sized subsets of the data. A model update is computed at the device level and communicated to a central server. A large number of these updates or gradients are combined at the central server during each iteration of training. A global update to the central model is computed as the average of local updates.

Although we can train the individual speaker model using only the client speech of a speaker on a given device, we also train the individual speaker model using client and impostor speech to make the model more robust. The performance results of these two approaches are described in the experimental results sub-section. Thus, as it is shown in Fig. 1, we use two different methods to generate impostor speech for each speaker on a device. The first method randomly selects the speech of other persons from Voxceleb dataset [10] as impostor speech for a given speaker. In the second method, we train a GAN model to generate impostor speech as it is not always easy to find speech samples of different persons in one's device. Thus, we use the work of [11] to train the GAN model. The GAN model is trained using the Voxceleb dataset. After the extraction of the impostor speech, they are used together with client speech to train an individual speaker model on each device.

The proposed system trains a deep neural network using distributed gradient descent across user-held training data on devices, using secure aggregation to protect the privacy of each user. Firstly, an individual model is trained locally on each individual device. Secondly, each individual device sends its locally trained models to the secure aggregator. The secure aggregator then builds a global model and sends this aggregated model to the main server. Finally, the main server sends the aggregated model to each individual device.

The secure aggregator is a class of secure multi-party computation algorithms wherein a group of mutually distrustful devices $d \in U$ each hold a private value x_u and collaborate to compute an aggregate value, such as the sum $\sum_{u \in U} x_u$, without revealing to one another any information about their private value except what is learnable from the aggregate value itself.

The proposed federated learning system for speaker recognition consists of two levels of optimization as it is shown in Algorithm 1. These are local optimization performed on participating clients, and a server step to update the global model. Algorithm 1 shows that the devices only communicate updated weights instead of speech data, which remain secure locally. Thus, this technique keeps the privacy of the user's speech data.

One of the main challenges of federated learning is the transfer of a large number of updated model parameters from the users to the server, whose throughput is typically constrained [12, 8, 13, 14]. This challenge can be tackled by reducing the number of participating users, for example, by scheduling policies [15, 16].

Algorithm 1 FedAvg. The *C* devices are indexed by *c*, epochs are indexed by *e*, and *n* is the number of speech samples.

Initialize w^0
for each *epoch* $e = 1, 2, ...$ **do**
  $D \leftarrow (random\ subset\ of\ M\ clients)$
  for each *client* $c \in C$ **do**
    $\hat{w}_c^e \leftarrow ClientUpdate(c, w^e)$
    $\Delta w_c^e = w^e - \hat{w}_c^e$
  end for
  $\bar{w}^e = \sum_{c=1}^{C} \frac{n_c}{n} \Delta w_c^e$ ▷ $weighted\ average$
end for
$w^{e+1} = w^e - \eta \bar{w}^e$ ▷ $Serverupdate$

3. Experiments

3.1. Experimental Setup

The input features of the system are mel-spectrograms computed within a 30 ms frame window at 10 ms shift using Librosa [17]. Mean and variance normalization is performed on every frequency bin of the spectrum. The mel-spectrogram features are extracted from the first 3.5 seconds of Voxceleb-1 audio files. Thus, the size of the mel-spectrogram is 350×80. Since the size of mel-spectrograms is 350 by 80, we use "Conv 2D" to train the model. The CNN architecture used in this work is similar to VGG-M [18], widely used for image classification and speech-related applications [19]. The details are shown in Table 1. We also apply a max pooling layer of size 2 by 2, batch normalization and dropout layers.

Our system has been implemented using the Keras deep learning library [20] to train the model. The network is trained on Titan X GPUs for 100 epochs or until the validation error stops decreasing, whichever is sooner, using a batch-size of 64. We use SGD with momentum (0.9), weight decay (5×10^{-4})

Table 1: *The architecture used for speaker verification*

Layer	Kernel	Filters	Output size
Conv-1	3 X 3	64	350 X 80 X 64
Conv-2	3 X 3	128	175 X 40 X 128
Conv-3	3 X 3	256	87 X 20 X 256
fc-1	-	1000	-
fc-2	-	400	-
fc-3	-	1	-

and a logarithmically decaying learning rate (initialised to 10^{-2} and decaying to 10^{-8}).

The proposed speaker verification system has been carried out on the VoxCeleb-1 database [10]. It contains 148 642 development and 4874 test utterances, which belong to 1211 and 40 speakers, respectively. We selected 25 speakers' speech data from the development set to train the proposed model. We used 80 % of each speaker data to train an individual speaker dependent model and used 20 % of the speaker data to do the evaluation. For example, if speaker 1 has 100 speech files in the development set of Voxceleb-1, we use 80 files to train the speaker model and 20 files to do the evaluation. We can not use the test set of Voxceleb-1 for this work as there is no overlap between the speakers in the development and test set.

Firstly, we train the individual speaker model using only the true client speech of a speaker on each device. However, since the number of files for each speaker are few in Voxceleb-1 dataset (most of the speakers have less than 100 speech files), training individual speaker models using only true client speech led to an over-fitting problem. Thus, we trained the individual speaker models using the true speech of the speaker and by using the speech of other speakers as impostor speech.

We use two different methods to generate the impostor speech for each individual device. In the first method, we take the speech of other speakers from the Voxceleb dataset as impostor speech. We select 900 samples of other persons as impostor speech for each speaker on a given device. In the second method, the impostor data is generated using the GAN model. We use the work of [11] to train the GAN model on Voxceleb dataset. Similar to the first method, we generate 900 impostor speech data for each individual device.

The main issue of using the GAN model to generate the impostor speech is its training time. The computational cost of training the GAN model using 50 hours of Voxceleb dataset on Quadro P2000 GPU is 3.5 hours. Once the GAN model is trained, the extraction of impostor speech samples on the edge devices is quite fast.

The performance of the proposed system is evaluated using equal error rate (EER) which is the rate at which both acceptance and rejection errors are equal.

3.2. Experimental Results

Figure 2 and 3 show the performance of the individual and aggreated speaker models. In the case of the individual model, a speaker model is first trained for each speaker using his/her own speech data. Then, the speech samples of the speaker is evaluated using this trained speaker model. In this work, we use the individual model as a baseline system (i.e, 25 individual speaker models are trained using the speech samples of speakers in 25 devices). But, in the case of the aggregated model, each of the 25 devices send their parameters to the secure aggregator and the aggregator takes the average of these parameters and sets

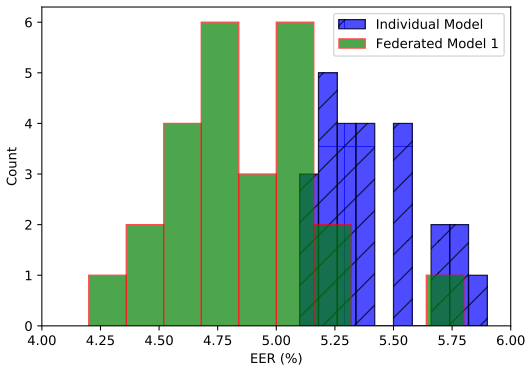

Figure 2: *Equal error rate (EER) of 25 devices using the individual vs federated model. The impostor speech samples are randomly selected from from Voxceleb dataset.*

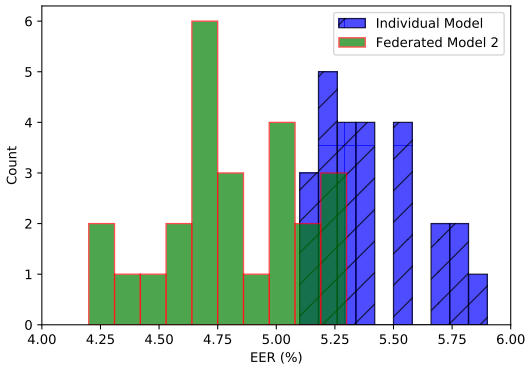

Figure 3: *Equal error rate (EER) of 25 devices using the individual vs federated model. The impostor speech samples are generated using GAN.*

them as its new weight parameters and passes them back to the 25 devices. Thus, we call this aggregated (federated) speaker model. The main difference between the models in Figure 2 and Figure 3 is in the generation of impostor speech. In the case of Fig. 2, the impostor speech samples are generated by selecting other persons' speech. But, in the case of Fig. 3, we use the GAN model to generate the impostor speech.

As shown in Fig. 2 and Fig. 3, the EER of all 25 devices when each device uses its own individual model is greater than 5.1. But, when we use the federated models, irrespective of impostor generation method, most of the devices provide us EER value of less 5.1. In the first method where the speech of other persons are used as impostor speech (i.e., Federated Model 1), 18 devices provide us EER value of less than 5.1 In the second aggregation method where we use GAN to generate impostor speech (i.e., Federated Model 2), 20 devices provide EER value of less than 5.1.

The box plots in Figure 4 depict the EER distribution of the 25 devices of the individual and federated models. The figure shows the minimum, lower quartile, median, upper quartile, and maximum EER values, respectively. As it is shown in the figure, the use of federated models, irrespective of impostor generation methods, provides better average EER than the individual models. While the average EER of the individual models 5.4%, federated model 1 provides us an average EER of 4.85%. This

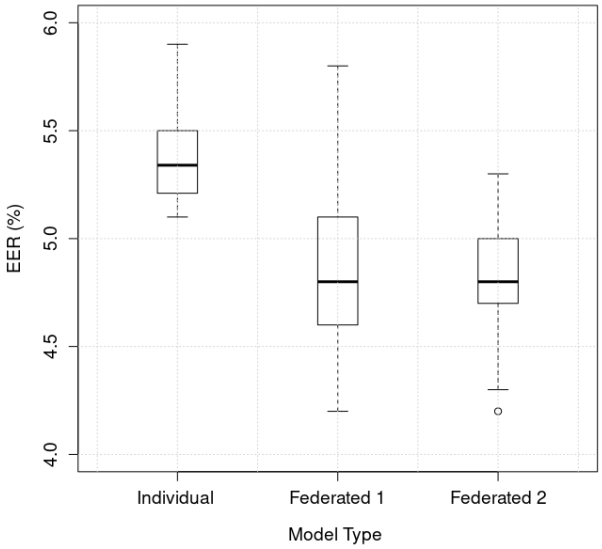

Figure 4: *Box plot of the equal error rates (EER) in individual and federated models of the 25 devices.*

represents a 10.19% relative EER improvement compared to the average of the individual models. Similarly, the use of federated model 2 provides us an average EER of 4.8%. This represents a 11.11% relative EER improvement compared to the average of the individual models.

We can further see that the range of EER values for federated model 1 is larger than the other two models, indicating that the model is less robust. The range of of federated model 2 has very little overlap with the individual model, indicating that model 2 gives a clear improvement in performance.

Figure 2 and 3 show that whether we use GAN or use other persons speech as impostor data, the aggregated speaker model provides better average EER than the individual model. The figures also reveal that the two aggregated methods provide more or less similar average EER. This shows that we can use GAN to generate impostor speech on individual devices rather than transferring impostor data from other sources to the individual devices.

The reported results thus demonstrate that federated learning is a useful tool for speaker recognition. The experimental results show that the aggregated models do not always provide better EER results compared to the individual model for all the devices, but the aggregated speaker models provide better average EER than the individual models. Out of 25 devices, the two aggregated models provide better average EER compared to the individual models on 23 devices.

In addition to the individual and federated models, we carried out another experiment by pooling all the speech samples from the 25 speakers and trained one generic speaker model on a single computer. However, the generic model faces an under-fitting problem because of shortage of data. Thus, the EER computed using the generic model on the test set is higher than both the individual and federated model. This shows that federated learning techniques can be used to alleviate the shortage of training data in speaker recognition applications.

Since the proposed work uses only 25 devices to compare the performance of the individual against the federated model, we use Student's t-test statistical technique to check if the means of individual and federated models are significantly different from each other. Student's t-test uses a null hypothesis and an alternate hypothesis. The null hypothesis is valid when all the sample means are equal, or they don't have any significant difference. Thus, we compared the EERs of the three experiments to make sure that the EER differences are because of the federated model, not merely by chance.

We compute two Student's t-test values. While the first one compares the individual model against the federated model that selects impostor speech from Voxceleb (i.e., federated model 1), the second one compares the individual model again the federated model that uses GAN to generate impostor speech (i.e., federated model 2). The Student's t-test computation on the first and second comparison provides P values of 0.0001 and 0.00001, respectively. Thus, since the P values in both cases are less than the standard significance level of 0.05, we reject the null hypothesis. Thus, the experimental results of Fig. 2 and 3 show that the difference of the mean EER values of the individual and federated models are statistically significant.

Finally, in the proposed work, each device sends its model updates, (i.e., the learned parameters) to the secure aggregator only once (i.e., each device sends the local model to the secure aggregator after it finishes the training). Thus, the federated model results reported in Fig. 2 and 3 use only one model update. We also assess the impact of updating the local model updates more than once. Thus, we run another experiment where each device sends its model updates two times to the secure aggregator. Our experimental analysis shows that updating the local model update more than once does not improve the EER. One possible reason could be since the training data used in each device is almost similar for each training phase, the EER results of updating the local models either once or twice are similar. The real performance could be revealed if the data been partitioned among devices, but we can not use this method since the privacy of the data would then be compromised.

4. Conclusions

Federated learning (FL) is a technique for protecting privacy by keeping speech data local to each participating edge device and sharing only model updates. In this work, we propose the use of federate learning for speaker recognition. The proposed technique is a decentralized training method that does not require devices to send their raw data to central servers. Instead, the users' data is stored and processed only on the corresponding edge device. Training is thus carried out only locally, and each device contributes updates to a central model. Then, the secure aggregator creates one federated model from the local models and distributes it through the main server back to the devices. The proposed system provides two main advantages. Firstly, since raw data does not leave the individual devices, the privacy of the speaker is retained. Secondly, the experimental results show that the aggregated model provide better average better equal error rate than the individual models. The experimental results also proved that the federated model provides a better average EER than the global model which is trained on a single server by pooling all the speech from 25 devices.

In future work, we should study better aggregation methods rather than using a simple averaging technique. In addition, we should also explore the effect of increasing the number of devices beyond the 25 devices used in this work.

5. Acknowledgment

This work has been supported by the Jane and Aatos Erkko foundation funding under contract 700795 AUTHSPKR.

6. References

[1] D. A. Reynolds, "Automatic speaker recognition using gaussian mixture speaker models," in *The Lincoln Laboratory Journal*. Citeseer, 1995.

[2] S. Furui, "An overview of speaker recognition technology," in *Automatic speech and speaker recognition*. Springer, 1996, pp. 31–56.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] J. Chen and X. Ran, "Deep learning with edge computing: A review." *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[5] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker *et al.*, "Large scale distributed deep networks," 2012.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[9] Y. Rahulamathavan, K. R. Sutharsini, I. G. Ray, R. Lu, and M. Rajarajan, "Privacy-preserving ivector-based speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 496–506, 2018.

[10] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[11] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *ICLR*, 2019.

[12] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," *ArXiv e-prints*, pp. arXiv–1602, 2016.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[14] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, 2020.

[15] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.

[16] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Update aware device scheduling for federated learning at the wireless edge," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2598–2603.

[17] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, J. Moore, D. Ellis, D. Repetto, P. Viktorin, J. F. Santos *et al.*, "Librosa: v0. 4.0," *Zenodo 2015*, 2015.

[18] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[19] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Asian conference on computer vision*. Springer, 2016, pp. 251–263.

[20] F. Chollet *et al.*, "Keras (2015)," 2017.