IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

GTH-UPM System for Albayzin Multimodal Diarization Challenge 2020

Cristina Luna-Jiménez[1], *Ricardo Kleinlein*[1], *Fernando Fernández-Martínez*[1], *José Manuel Pardo-Muñoz*[1], *José Manuel Moya-Fernández*[1]

[1]Information Processing and Telecommunications Center, E.T.S.I. de Telecomunicación, Universidad Politécnica de Madrid, 28040, Madrid, Spain

cristina.lunaj@upm.es, ricardo.kleinlein@upm.es, fernando.fernandezm@upm.es, josemanuel.pardom@upm.es, jm.moya@upm.es

Abstract

This paper describes the multimodal diarization system proposed by the GTH-UPM team to Albayzin Multimodal Diarization Challenge 2020. The submitted solution consists of 2 separate diarization systems that work on visual and aural components.

The visual diarization solution exploits web resources, as well as provided enrollment images. First, these images feed a facial detector. Next, all the discovered faces are introduced into FaceNet to generate embeddings. After this, we apply a clustering algorithm on extracted embeddings, obtaining a representative cluster for each participant. Each centroid of the representative clusters acts as a participant model. When a new embedding extracted from a facial image of the program arrives at the system, it receives the label that corresponds to the closest centroid identity among all the given participants, as long as it exceeds a fixed quality threshold.

The aural speaker diarization problem is tackled as a classification task, in which a deep learning model learns the mapping between automatically-extracted sequences of aural x-vectors and speaker identities. These sequences aid in overcoming the scarcity of training samples per speaker.

The best results sent reached a DER of 66.94% for visual diarization and a DER of 125.24% for aural diarization on the test set.

Index Terms: Multimodal diarization, clustering, biLSTM, attention

1. Introduction

Speaker diarization systems aim to identify who is talking and when [1]. Similar to speaker diarization, visual diarization systems try to discover who is in the scene and when. Although visual diarization is not as common as speaker diarization, there are already some publications that start to include this modality to reduce the diarization error rate [2][3]. Evidence of this tendency change is the Albayzin Multimodal Diarization Challenge at IberSPEECH conference [4].

Previous year approaches [5][6][7] address visual and aural diarization in different ways, but all of them include some common modules. These modules consist of a tracker and a detector of participants or speech, a feature extractor to generate embeddings, and an identity recognizer. In line with prior work, we propose a visual diarization system with mentioned modules, including information extracted from a novel source, Google Images.

Although traditionally speaker diarization has been tackled from an unsupervised point of view [8, 1], recent research seems to successfully approach the problem from a supervised perspective by means of recurrent neural models [9]. Additionally, attention mechanisms have lately enabled great improvements in many fields, including speaker diarization [10]. On a similar basis, x-vectors were proposed by Snyder *et al.* as a way to map variable-length utterances to fixed-dimensional embeddings that greatly enhance speaker's acoustic characterization [11]. Therefore, we propose a speaker diarization system based on the supervised learning of a map between fixed-length sequences of x-vectors and speaker identities via recurrent and attention neural models.

The structure of the paper is as follows: Section 2 describes the proposed diarization system. Section 3 summarises the computational cost of the experiments. Section 4 presents the main experiments performed and the results obtained for development and test sets. Finally, in Section 5, we illustrate the extracted conclusions and future research work.

2. System description

In this section, we present the visual and aural diarization systems. The visual diarization system bases its functionality on a weakly-supervised strategy, solving the diarization and the attribution tasks. Regarding speaker diarization, we employ a fully-supervised model, trained on the development programs.

In what follows, we explain the different components that constitute them and their relationships to detect who is speaking or appearing in each program.

2.1. Visual Diarization Pipeline

Discerning identities from visual information still presents several challenges intrinsic to videos, like dealing with occlusions, blurring, etc. Our proposed solution overcome some of these challenges through four main modules: data acquisition, facial detection, identity recognition, and post-processing. A whole picture of the pipeline is in Fig. 1.

To homogenize the programs and accelerate the experiments, we worked at 5 fps with a resolution of 1.024 x 576.

2.1.1. Data acquisition

Due to the limited amount of images provided in the enrollment set, we included data acquired from Google Images. To afford it, we developed a tool to download these resources automatically. One inconvenience of this methodology is the uncertainty about the picture's content, i.e. whether downloaded images belong to the queried identity. To deal with noisy images and discard them, we apply a filtering process. This process consists of applying a face detector and a face recognizer over all the available material.

10.21437/IberSPEECH.2021-15

2.1.2. *Face detection*

Both images downloaded from Google and those provided in the enrollment set are passed through MTCNN [12]. MTCNN is a state-of-the-art face detector that returns face positions and their probability of being faces. In our settings, we fixed acceptance confidence of 98% and minimum face size of 80x80. Detected faces that do not accomplish these thresholds are removed.

2.1.3. *Face recognition*

To perform facial recognition, firstly, we extract embeddings from a pre-trained network, secondly, we make a clustering with these embeddings and, finally, we build a model per participant based on clustering results.

For the embeddings generation, we re-scale detected faces by MTCNN to a size of 160x160. These resized facial images are injected into FaceNet [13]. FaceNet is a state-of-the-art pretrained model in identity recognition. From each facial image introduced, FaceNet returns a 128-dimensional vector. This embedding represents the face in a latent space. Face embeddings corresponding to the same identity are closer to each other in the latent space than those corresponding to different identities.

As commented before, Google Images could return unexpected results. To remove the non-useful material, we apply a clustering algorithm. DBSCAN [14], from sklearn library [15], is the clustering algorithm selected since it does not require to set the desired number of clusters beforehand.

To detect the optimal working point of the participant clustering, we run several DBSCAN [14] instances, varying the epsilon value. More specifically, we scan epsilons between 1 and 12, in steps of 0.5. The rest of the DBSCAN parameters maintain their default value, except for the min_samples, fixed to 5. Once the scanning finishes, the optimal instance agrees with the maximum silhouette [16] coefficient obtained in the scanning. The silhouette coefficient is an unsupervised metric that measures the quality of each DBSCAN instance in terms of inter and intra-cluster distances.

Having the optimal DBSCAN [14] instance, we obtain the representative cluster of each identity by using this score:

$$SC(E_{CLT_i}, E_{ENR_j}) = w * E_{ENR_j} + (1 - w) * E_{CLT_i} \quad (1)$$

where E_{CLT_i} represents the percentage of embeddings in cluster i over the total number of embeddings of the participant j; E_{ENR_j} is the percentage of images of the enrollment set of participant j that belong to cluster i; and w is the weight that balance the contribution of each term.

By varying the equation's weight, we can increase or reduce the contribution of the enrollment images. A weight equal to 0 selects the cluster with the maximum number of embeddings as the representative. However, with a weight equal to 1, the algorithm chooses the cluster that contains more embeddings of the enrollment set as the representative, independently if it is the densest or not.

As the competition provides the enrollment images, they probably contain the face of the participant and can be used as a reference to know whether the cluster is of the expected participant or not. If the used images would be the images obtained from Google exclusively, the predominant cluster might not coincide with the expected participant since Google is noisy. For this reason, we compare the performance of the algorithm with two weights: 0 and 0.9 to study the effect of using the enrollment images as a guide against not using it, relying only on Google data. See Table 1.

We repeat this process with each participant. Thus, for each participant, we discard wrong downloaded pictures and select his/her representative cluster.

Finally, the character's model of each participant is the centroid of the representative cluster, i.e. the average vector given by all the embeddings that constitute its representative cluster.

2.1.4. *Post-Processing*

Intending to improve results, we included in the pipeline several post-processing techniques to refine results.

When a new face arrives at the system, the system calculates the cosine similarity between the embedding of the face and the key-participants models, i.e. the centroid of the representative cluster. The new face receives the label of the closest key-participant centroid, i.e. the key-participant with the highest cosine similarity. To reduce the false-positive rate, we fixed a quality threshold of 0.5. This quality filter lets modify the label from a key-participant to 'unknown' whenever its cosine similarity is lower than 0.5. Check Prediction & Evaluation box in Fig. 1.

To deal with occlusions and blurring in some frames, we applied a tracking process to extend predictions from a single frame to a track. We apply tracking in 2 steps:

1. Detect scenes: FFMPEG implements a scene detector. This detector applies the sum of absolute differences and returns a probability of scene change in each frame. When the frames' probability overpasses a certain threshold, it marks the start of a new scene. In the experiments, we use this detector with a threshold of 0.2.
2. Full-body tracking: we track people in each scene using a multi-person tracker [17]. This repository makes use of several pre-trained classification models able to detect people on an image. In our case, we selected the YOLO model and a people detection threshold of 0.7. The algorithm performs tracking by distinguishing overlapping bounding boxes in consecutive frames.

To match full-body bounding boxes, obtained in tracking, with facial bounding boxes, detected by MTCNN [12], we calculate the intersection over union (IoU) of all the pairs. The pair of full-body and facial bounding boxes that reaches the maximum IoU is matched. Once, there is a match between the face and the tracked body, it is possible to assign a single identity to the whole track, correcting face miss-detections and possible wrong predictions. The assigned identity to the whole track is the maximum voted, or the most times predicted along the faces that conform the track.

2.1.5. *Prediction of identity*

By passing new frames through all the modules, we extract the diarization file with all the participants' predicted in each moment. From this result, we generate a file called 'face pooling', that contains the list of participants detected by the visual diarization system at least once during the program. This reduced list of participants is a new source of information that the aural diarization applies to fine-tune the model.

2.2. Aural Diarization Pipeline

Speaker diarization systems aim at identifying who is talking and the timestamps that define such talking turn. Our proposed method treats the problem in the first stage as a supervised classification task, matching chunks of x-vectors with speaker

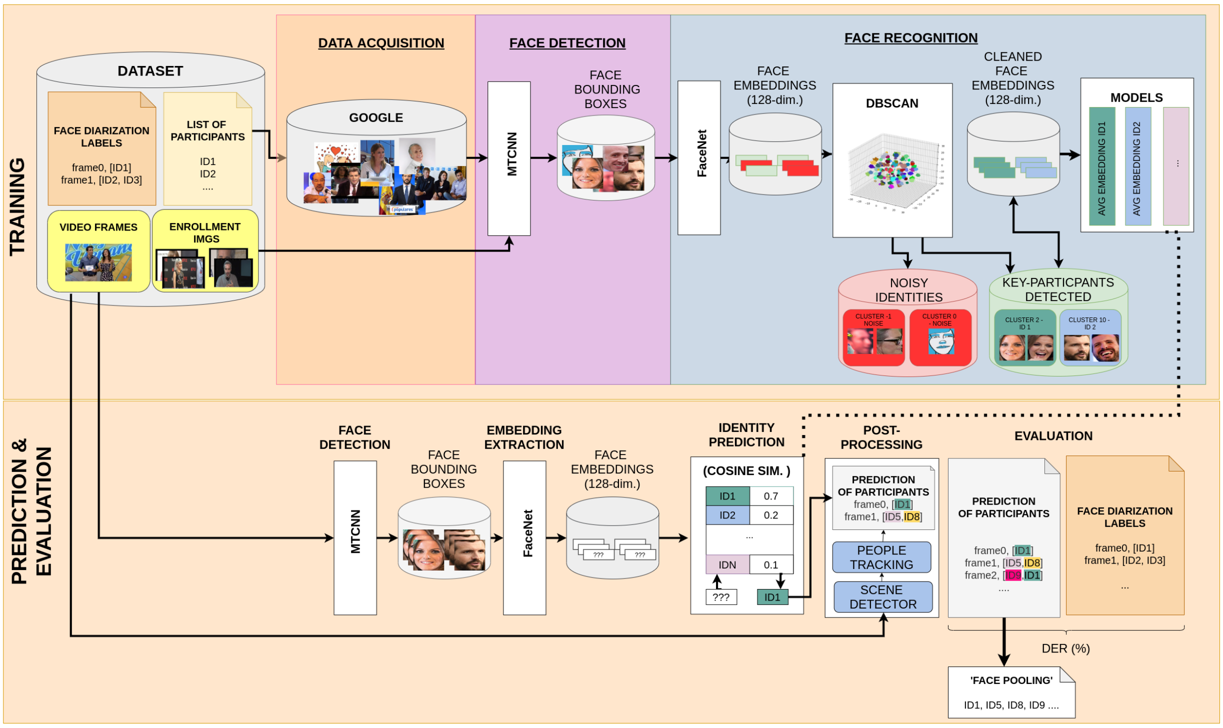

Figure 1: *Visual diarization pipeline*

names on the audios provided in the development set and the chunks of the enrollement set. Secondly, the temporal alignment of the samples let turn a classification problem into a speaker diarization estimation.

2.2.1. *Extraction of aural embeddings*

First of all, audio files originally provided in AAC format are converted into 16KHz, single-channel WAV format using the FFMPEG toolbox [18]. Next, we compute 512-dimensional aural x-vectors for every audio file following the default procedure explained in the III DIHARD competition [19]. We extract x-vectors every 0.75 seconds, spanning 1.5 seconds with 50% overlap between adjacent vectors. This procedure is applied over development, enrollment, and test data.

2.2.2. *Speaker Classification*

We desire to estimate a speaker name for every x-vector passed as input to the system. However, rather than a single x-vector, the predictive model's input is extended to include 5 consecutive samples. A sliding window extracts these samples by taking the current x-vector and its near context. Despite receiving a sequence of x-vectors as input, the outcome is a single prediction. This prediction is associated with the x-vector that occupies the middle position of the input sequence.

The classifier's architecture has an initial biLSTM layer with 50 units, followed by a self-attention layer and a fully-connected layer. In the end, a softmax activation layer provides speaker probabilities.

The classifier minimizes the cross-entropy loss using an Adam optimizer [20]. Both biLSTM and the attention layers have a dropout rate of 0.3. The initial learning rate is 0.001. We also include an early-stopping regularization that stops the model's training after 5 epochs without increasing the F1 classification score. To increase the robustness of the model from epoch to epoch, we implement a data augmentation technique. This approach adds white noise to the input embedding sequences at a random probability rate of 0.15.

For the system submitted as primary (p in Table 1), the learning process consists of two folds: First, we teach the base model (c1 in Table 1) to classify among all the potential identities given in development, test, and enrollment data, until convergence.

For training this model, we use the audios provided in the development set, that consist of the programs' records and the clips of the enrollment folder. To include the participants' voices that appear in the test set but not in the development set, we also add the clips of the test enrollment folder.

Once the base model is trained, we adapt it to each program by applying fine-tuning with the identities in the "face pooling" returned by the visual diarization branch.

To fine-tune the model, we get all the embeddings in the training set that include a participant of the 'face pooling' and re-train the model with this reduced set of participants, maintaining the previously mentioned configuration.

Therefore we end up having a fine-tuned and specialized model for every test program.

2.2.3. *Speech/Non-speech Segmentation*

Voice Activity Detectors (VADs) are a customary solution to discern non-speech parts of the audio from actual speech. We employ the default DIHARD's VAD module over the test partition data [19]. This step enables to remove the parts predicted as non-speech. This filtering allows refining our predictions to a greater degree of detail.

In environments such as TV broadcast debates, in which talking turns can change in a matter of milliseconds due to interruptions, the use of a VAD-based speech filter gains importance.

3. Computational Cost

Visual diarization experiments were carried out in a computer with an Intel® Core™ i7-3770K Processor, 32GB RAM and a GPU NVIDIA Titan X Pascal, 12 GB. The most time-consuming tasks were the FaceNet embeddings extraction and the tracks calculation. In total, the version without tracking and no-parallel processing takes $xRT \approx 1.18$.

Regarding the aural component, the overall expense in time account up to 2 hours and a half approximately. We used a Nvidia GeForce RTX 2070 with 8Gb of RAM memory. The rest of computer specifications are the same as described for the visual part.

4. Experiments and Results

Table 1 collects the best results obtained in each modality. The best system submitted to the competition in the visual modality for the development set (the second row in Table 1) reached a DER of 66.94% on the test set. This system performs cluster assignment with a weight of 0.9, giving more relevance to the number of enrollment images. Additionally, it includes tracking as post-processing.

The first system in Table 1 is the same as the previously mentioned without including the tracking module. As we can see, the tracking strategy seems to improve the final Diarization Error Rate in the development set, but not in the test set. This mismatch-effect is explained by the different types of programs that define the test set, an effect that our tracking solution can not address since it uses the same parameters for all the programs. To confirm this behavior, we repeated the experiments adapting the tracking parameters to the test set, and although the DER of the test set decreased (60.32%), the DER in the development set increased (73.93%).

After analyzing the results per program for the tracking configuration of Table 1 against the non-tracking version, we detect that DER in the test set for programs with acronym AT, BR, NFMY, and WU decreases in 4.97%, 7,42%, 1.24%, and 11.20%, respectively, using tracking module; meanwhile for programs BN, CA, EP, LD, ML, and SFT the DER increases in 5.55%, 38.74%, 1.04%, 4.41%, 157.2%, and 9.42%, respectively, when we use the tracking module. Notice that these values are the average per type of program, not considering durations.

We performed additional experiments not shown in Table 1, using only Google Images (DER = 76.55%) and using only enrollment images (DER = 75.41%) with w=0.0 and without tracking. Experiments reveal that joining both sources of information can improve diarization results (DER = 75.24 %), although the difference is not too significant since the model uses the average of facial embeddings that softens the effect of having more images.

Regarding aural diarization, it seems that the system based on ‘face pooling’ performs worse than the version without fine-tuning the base model. Uniquely, programs of type BN, CA, NFMY, SFT, and WU improves DER values. These results reveal that in spite of the reduction in the number of input identities to the model (40%), it is not enough to compensate the 7% of lost identities. Furthermore, it is essential to mention that the model employed in speaker diarization is fully-supervised, which means that it is tailored to the training domain. As a result, when it faces different programs or lack of data of some participants, it reduces its performance, as has happened in this challenge.

Table 1: *Diarization Error Rate for visual and aural systems*

System	DER dev. (18 IDs)	DER test (161 IDs)	Run
1. Visual Diarization w=0.9 - No Tracking	74.34 %	61.58 %	-
2. Visual Diarization w=0.9 - Tracking	**59.63 %**	**66.94 %**	**p**
3. Aural Diarization without ‘face pooling’	**-**	**125.24 %**	**c1**
4. Aural Diarization with ‘face pooling’	-	131.59 %	p

5. Conclusion and future work

In this paper, we present the GTH-UPM systems employed for solving the 2020 Albayzin Multimodal Diarization Challenge. The visual recognition solution is based in a weakly-supervised strategy that employs web resources, clustering and distances to obtain the participants that appear in each scene. Additionally, it also implements tracking and post-processing techniques to improve overall performance. To combine aural and visual diarization, we extract the participants detected in the visual diarization to fine-tune a biLSTM model with an attention mechanism.

Although results encourage us to continue with this research line, there are still some open issues to address in the future. One of the most important is how to exploit the images acquired from Google to improve visual diarization. We plan to test a more complex model rather than clustering to perform facial recognition and enhance the tracking module to automatically adapt its parameters to the type of program.

Regarding aural speaker diarization, we plan to explore new alternatives to increase the amount of available data to train the classifier model. Among such, we contemplate incorporating additional data augmentation techniques or, as we did in the visual diarization pipeline, recovering material from a second source of information.

Concerning fusion, we also consider investing some effort in testing other strategies to improve global DER combining embeddings of both modalities.

To conclude, this paper contributes to web resources’ exploitation and the study of diarization systems of two (visual and aural) modalities.

6. Acknowledgements

The work leading to these results has been supported by the Spanish Ministry of Economy, Industry and Competitiveness through the CAVIAR (MINECO, TEC2017-84593-C2-1-R) and AMIC (MINECO, TIN2017-85854-C4-4-R) projects (AEI/FEDER, UE).

Ricardo Kleinlein’s research was supported by the Spanish Ministry of Education (FPI grant PRE2018-083225).

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

7. References

[1] I. Viñals, A. Ortega, J. Villalba, A. Miguel, and E. Lleida, "Domain adaptation of PLDA models in broadcast diarization by means of unsupervised speaker clustering," in *Proc. Interspeech 2017*, 2017, pp. 2829–2833. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-84

[2] P. Campr, M. Kunešová, J. Vaněk, J. Čech, and J. Psutka, "Audio-video speaker diarization for unsupervised speaker and face model creation," in *Text, Speech and Dialogue*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Cham: Springer International Publishing, 2014, pp. 465–472.

[3] P. A. Marín-Reyes, J. Lorenzo-Navarro, M. Castrillón-Santana, and E. Sánchez-Nielsen, "Who is really talking? a visual-based speaker diarization strategy," in *Computer Aided Systems Theory – EUROCAST 2017*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, Eds. Cham: Springer International Publishing, 2018, pp. 322–329.

[4] J. Luque, A. Bonafonte, F. A. Pujol, and A. J. S. Teixeira, Eds., *Fourth International Conference, IberSPEECH 2018, Barcelona, Spain, 21-23 November 2018, Proceedings*. ISCA, 2018. [Online]. Available: https://doi.org/10.21437/IberSPEECH.2018

[5] B. Maurice, H. Bredin, R. Yin, J. Patino, H. Delgado, C. Barras, N. W. D. Evans, and C. Guinaudeau, "ODESSA/PLUMCOT at Albayzin Multimodal Diarization Challenge 2018," in *Fourth International Conference, IberSPEECH 2018, Barcelona, Spain, 21-23 November 2018, Proceedings*, J. Luque, A. Bonafonte, F. A. Pujol, and A. J. S. Teixeira, Eds. ISCA, 2018, pp. 194–198. [Online]. Available: https://doi.org/10.21437/IberSPEECH.2018-39

[6] M. A. I. Massana, I. Sagastiberri, P. Palau, E. Sayrol, J. R. Morros, and J. Hernando, "UPC multimodal speaker diarization system for the 2018 Albayzin Challenge," in *Fourth International Conference, IberSPEECH 2018, Barcelona, Spain, 21-23 November 2018, Proceedings*, J. Luque, A. Bonafonte, F. A. Pujol, and A. J. S. Teixeira, Eds. ISCA, 2018, pp. 199–203. [Online]. Available: https://doi.org/10.21437/IberSPEECH.2018-40

[7] E. Ramos-Muguerza, L. D. Fernández, and J. L. Alba-Castro, "The GTM-UVIGO system for audiovisual diarization," in *Fourth International Conference, IberSPEECH 2018, Barcelona, Spain, 21-23 November 2018, Proceedings*, J. Luque, A. Bonafonte, F. A. Pujol, and A. J. S. Teixeira, Eds. ISCA, 2018, pp. 204–207. [Online]. Available: https://doi.org/10.21437/IberSPEECH.2018-41

[8] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, 2003, pp. 411–416.

[9] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6301–6305, 2019.

[10] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 296–303, 2019.

[11] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[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, p. 1499–1503, Oct 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603342

[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.2015.7298682

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[16] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, p. 53–65, Nov. 1987. [Online]. Available: https://doi.org/10.1016/0377-0427(87)90125-7

[17] M. Kocabas and C. Heinrich, "Simple multi person tracker," 12 2019. [Online]. Available: https://github.com/mkocabas/multi-person-tracker

[18] S. Tomar, "Converting video formats with ffmpeg," *Linux Journal*, vol. 2006, p. 10, 2006.

[19] N. Ryant, K. W. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "Third DIHARD Challenge Evaluation Plan," *ArXiv*, vol. abs/2006.05815, 2020. [Online]. Available: https://arxiv.org/pdf/2006.05815.pdf

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980