Identification of Subjects Wearing a Surgical Mask from their Speech by means of x-vectors and Fisher Vectors⋆

José Vicente Egas-López[1] and Gábor Gosztolya[1,2]

[1] University of Szeged, Institute of Informatics, Szeged, Hungary
[2] ELRN-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
`{ egasj, ggabor } @ inf.u-szeged.hu`

Abstract. This paper addresses the classification of speech recorded from subjects while wearing a surgical mask. Here, we employ two different types of feature extraction methods: the x-vectors embeddings, which is the current state-of-the-art approach for Speaker Recognition; and the Fisher Vector (FV), that is a method originally intended for Image Recognition, but here employed to discriminate utterances. These approaches make use of distinct frame-level representations: MFCC and PLP. Using Support Vector Machines (SVM) as the classifier, we perform a technical comparison between the performances of the FV encodings and the x-vector embeddings for this particular classification task. We find that the Fisher vector encodings provide better representations of the utterances than the x-vectors do for this specific dataset. Moreover, we show that a fusion of our best configurations outperforms all the original baseline scores.

Keywords: speech recognition · computational paralinguistics · fisher vectors, x-vectors · surgical mask

1 Introduction

The Computational Paralinguistics differs from Automatic Speech Recognition in that the latter seeks to determine the *content* of the speech of an utterance, while the former seeks to understand the *way* that the speech is spoken. There are different types of techniques that attempt to solve this problem in Computational Paralinguistics. Methods such as the i-vector Approach, the Fisher vector,

neural networks, among others, are being increasingly used by researchers to address paralinguistic issues. This can be seen in studies like diagnosing neurodegenerative diseases using the speech of the patients [1,2,3]; the discrimination of crying sounds and heartbeats [4]; or the estimation of the sincerity of apologies [5]. These studies aim to distinguish the latent patterns existing within the speech of a subject and not the content of it.

The INTERSPEECH ComParE Challenge, annually organized since 2009 [6], has provided a wide variety of Computational Paralinguistics problems each year. These types of challenges seem to encourage its participants to use or devise state-of-the-art techniques to handle the states and characteristics latent in an audio signal. This year, the challenge offers three tasks; but here we will just focus on one of them, namely, the *Mask* Sub-Challenge.

The 2020's challenge involved speech recordings of German speakers while wearing a surgical mask, and also while not wearing one. The task was to determine whether the utterance corresponds to a speaker whose speech was recorded while wearing the mask or not. The baseline reported by the organizers is an UAR (Unweighted Average Recall) score of 70.8%, which corresponds to a non-fused score. And a 71.8% for the fusion of the best four configurations for the *Mask* Sub-Challenge. Forensics and *'live'* communication between surgeons may benefit from a system that could determine whether a subject is wearing a mask based on their speech [7].

Lots of speaker recognition systems these days are based on i-vectors [8]. The i-vector system utilizes a GMM-UBM (Universal Background Model) to extract a fixed-dimension feature called *i-vector*. This is a robust technique that was and still is the state-of-the-art for many speaker recognition/verification approaches [9,10]. Also, i-vectors have been used in computational paralinguistics and offer promising results when assessing Alzheimer's from speech [11], or at the moment of classifying depressed speech [12]. Nonetheless, there are more meaningful features that seem to provide better representations of frame-level features than the i-vectors do.

Embeddings extracted from a Feed-Forward Deep Neural Network are gradually replacing i-vectors; such embeddings are called *x-vectors*. Regarded as the new state-of-the-art technique for speaker recognition systems [13], x-vectors can capture meta-information such as the gender of the speaker, as well as their speech rate (i.e. long-term speech traits). Researchers are increasingly using such representations in their studies, especially in text-independent approaches (see e.g. [14,15,16,17]). Also, *x-vectors* have already been applied to paralinguistics; studies like [18,19,20] reported high performances at classifying emotions, Alzheimer's, or age and gender of subjects.

As a contribution to the ComParE Challenge, here, we perform the chosen task via two different methodologies. The Fisher Vector (FV) approach [21], which is an encoding method originally developed to represent images as gradients of a global generative GMM of low-level image descriptors; mainly used in image recognition [22]. And we also employ the DNN embeddings approach (i.e.
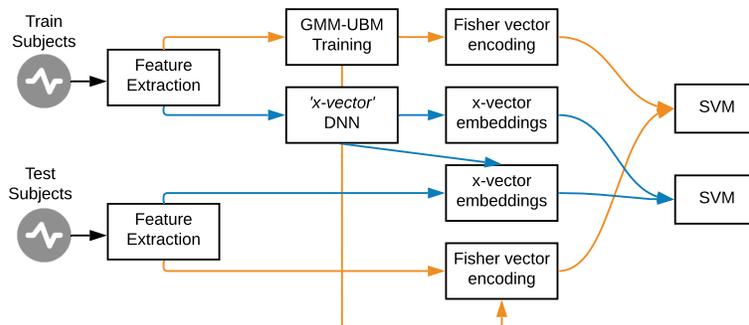
Fig. 1. The generic methodology applied in this study.

x-vector system) where the role of the DNN is to perform a mapping between variable-length utterances and fixed-dimensional embeddings.

The workflow proposed is the following. First, we use two types of frame-level representations, i.e., MFCCs and PLPs extracted from the audio signals. Second, we process the frame-level information obtained utilizing two different techniques: the FV and the x-vector approaches. And third, we classify and evaluate FV and x-vector features individually. Finally, we opt for a late-fusion of the best configurations.

2   Data

The Mask Augsburg Speech Corpus (MASC) comprises recordings of 32 German native speakers. It has a total duration of 10 h 9 min 14 sec; segmented into chunks of 1 sec. The recordings have a rate of 16 kHz. The total number of utterances is 36554: 10895 for train, 14647 for development, and 11012 for test. The subjects were asked to perform specific types of tasks and recorded their speech while wearing and not wearing a surgical mask. (See more details in [7]).

3   Feature Extraction and Evaluation Methods

As depicted in Figure 1, the steps carried out in our study are as follows: (1) Feature extraction (MFCCs and PLP); (2a) Train GMM-UBM using utterances from the training set, (2b) Train the DNN for the x-vectors utilizing the training set and its *augmented* version; (3a) Extract Fisher vector features from the datasets employing the GMM-UBM model, (3b) Extract embeddings from the DNN; and, (4a/4b) Independently classify the FV and x-vectors representations using SVM.

Table 1. DNN architecture of the x-vector system. It comprises five frame-level layers, a statistics pooling layer, two segment-layers and a final softmax layer as output. N represents the number of training speakers in the softmax layer. The DNN structure here is the same as that shown in Snyder et al. [24].

Layer	Layer context	Tot. context	In, Out
frame1	[t-2, t+2]	5	120, 512
frame2	{t-2, t, t+2}	9	1536, 512
frame3	{t-3, t, t+3}	15	1536, 512
frame4	{t}	15	512, 512
frame5	{t}	15	512, 1500
stats pooling	[0, T}	T	1500T, 3000
segment6	{0}	T	3000, 512
segment7	{0}	T	512, 512
softmax	{0}	T	512, N

3.1   Frame-level features

Here, we used the well-known Mel-Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictions (PLP) frame-level representations. Both have 13 dimensions, a frame-length of 25ms and a sliding window of 3ms. Moreover, since x-vectors are extracted from a DNN, an additional configuration for the MFCCs called *high-resolution (hires)* was utilized. This allows us to maintain all the cepstra while decorrelating the MFCCs. The MFCC-hires configuration is intended for neural network training. This configuration has the same values as those previously described, except that it extracts 40 cepstral coefficients, the number of mel-bins is 40, and the low and high cut-off frequencies are 20 and -400, respectively (see e.g. in [23]). Also, non-speech frames are removed from all the representations employing VAD.

3.2   x-vectors

The x-vector approach can be thought as of a neural network feature extraction technique that provides fixed-dimensional embeddings corresponding to variable-length utterances. Such a system can be viewed as a feed-forward Deep Neural Network (DNN) that computes such embeddings. Below, we will describe the architecture of the DNN (based on [13]) and the embeddings that are extracted from it.

DNN structure Table 1 outlines the architecture of the DNN. The *frame-level* layers have a time-delay architecture, and let us assume that t is the actual time step. At the input, the frames are spliced together; namely, the input to the current layer is the spliced output of the previous layer (i.e. input to layer *frame3* is the spliced output of layer *frame2*, at frames $t-3$ and $t+3$). Next,

the *stats pooling* layer gets the T frame-level output from the last frame-level layer (*frame5*), aggregates over the input segment, and computes the mean and standard deviation. The mean and the standard deviation are concatenated and used as input for the next *segment* layers; from any of these layers the *x-vectors* embeddings can be extracted. And finally, the *softmax* output layer (which is discarded after training the DNN) [13,24,25].

Instead of predicting frames, the network is trained to predict speakers from variable-length utterances. Namely, it is trained to classify the N speakers present in the train set utilizing a multi-class cross entropy objective function (see Eq. 1). Let K be the number of speakers in N training segments. Then, the probability of the speaker k given T input frames $(x_1^{(n)}, x_2^{(n)}, ..., x_{1:T}^{(n)})$ is given by: $P(spkr_k|x_{1:T}^{(n)})$. If the speaker label for segment n is k, then the quantity of d_{nk} is 1, and 0 otherwise [24].

$$E = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \ln P(spkr_k|x_{1:T}^{(n)}). \tag{1}$$

Embeddings The embeddings produced by the network described above capture information from the speakers over the whole audio-signal. Such embeddings are called *x-vectors* and they can be extracted from any *segment* layer; that is, either *segment6* or *segment7* layers (see Table 1). Normally, embeddings from the *segment6* layer give a better performance than those from *segment7* [13]. In this study, these type of representations can capture meaningful information from each utterance. This embedding may help us to discriminate better the utterances due to the fact that the characteristics are acquired at the utterance level rather than at the frame-level. For this, we used the Kaldi Toolkit [26].

3.3   Fisher Vectors

The Fisher Vector approach is an image representation that pools local image descriptors (e.g. SIFT, describing occurrences of rotation- and scale-invariant primitives [27]). In contrast with the Bag-of-Visual-Words (BoV, [28]) technique, it assigns a local descriptor to elements in a visual dictionary, obtained via a Gaussian Mixture Model for FV. Nevertheless, instead of just storing visual word occurrences, these representations take into account the difference between dictionary elements and pooled local features, and they store their statistics. A nice advantage of the FV representation is that, regardless of the number of local features (i.e. SIFT), it extracts a *fixed-sized* feature representation from each image.

The FV technique has been shown to be quite promising in image representation [21]. Despite the fact that just a handful of studies use FV in speech processing (e.g. for categorizing audio-signals as speech, music and others [29], for speaker verification [30,31], and for determining the food type from eating sounds [32]), we think that FV can be harnessed to improve classification performance in audio processing.

Fisher Kernel The Fisher Kernel (FK) seeks to measure the similarity of two objects from a parametric generative model of the data (X) which is defined as the gradient of the log-likelihood of X:

$$G_\lambda^X = \bigtriangledown_\lambda \log \upsilon_\lambda(X), \tag{2}$$

where $X = \{x_t, t = 1, \ldots, T\}$ is a sample of T observations $x_t \in \mathcal{X}$, υ represents a probability density function that models the generative process of the elements in $\mathcal{X}$ and $\lambda = [\lambda_1, \ldots, \lambda_M]' \in R^M$ stands for the parameter vector υ_λ [33]. Thus, such a gradient describes the way the parameter υ_λ should be changed in order to best fit the data X. A novel way to measure the similarity between two points X and Y by means of the FK can be expressed as follows [21]:

$$K_{FK}(X, Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y. \tag{3}$$

Since F_λ is positive semi-definite, $F_\lambda = F_\lambda^{-1}$. Eq. (4) shows how the Cholesky decomposition $F_\lambda^{-1} = L_\lambda' L_\lambda$ can be utilized to rewrite the Eq. (3) in terms of the dot product:

$$K_{FK}(X, Y) = \mathscr{G}_\lambda^{X\prime} \mathscr{G}_\lambda^Y, \tag{4}$$

where

$$\mathscr{G}_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X). \tag{5}$$

Such a normalized gradient vector is the so-called *Fisher Vector* of X [33]. Both the FV $\mathscr{G}_\lambda^X$ and the gradient vector G_λ^X have the same dimension.

Fisher Vectors Let $X = \{X_t, t = 1 \ldots T\}$ be the set of D-dimensional local SIFT descriptors extracted from an image and let the assumption of independent samples hold, then Eq. (5) becomes:

$$\mathscr{G}_\lambda^X = \sum_{t=1}^{T} L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X_t). \tag{6}$$

The assumption of independence permits the FV to become a sum of normalized gradients statistics $L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(x_t)$ calculated for each SIFT descriptor. That is:

$$X_t \to \varphi_{FK}(X_t) = L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X_t), \tag{7}$$

which describes an operation that can be thought of as a higher dimensional space embedding of the local descriptors X_t.

The FV extracts low-level local patch descriptors from the audio-signal spectrogram. Then, a GMM with diagonal covariances models the distribution of the extracted features. The log-likelihood gradients of the features modeled by the parameters of such GMM are encoded through the FV [33]. This type of encoding stores the mean and covariance deviation vectors of the components k that form the GMM together with the elements of the local feature descriptors. The utterance is represented by the concatenation of all the mean and the covariance vectors that gives a final vector of length $(2D + 1)N$, for N quantization cells and D dimensional descriptors [33,34]. Here, we use FV features to encode the MFCC features extracted from the audio-signals of the *Mask* dataset.

3.4   Support Vector Machines (SVM)

A linear-SVM classifier was utilized to discriminate the audio-signals. This algorithm was found to be robust even with a large number of dimensions and it was shown to be efficient when used with FV [33,35] due to it being a discriminative classifier that provides a flexible decision boundary. We used the libSVM implementation [36] with a linear kernel. As stated in the paper on this year's challenge [7], since 2009 (and also for this year), Unweighted Average Recall (UAR) has been the chosen metric for evaluating the performance of the classifiers.

4   Experimental setup

As for the Fisher vectors, the number of K GMM components utilized to compute the FVs ranged from $2, 4, 8$, to 512. The construction of the FV encoding was performed using a Python-wrapped version of the VLFeat library [37]. Both MFCC and PLP representations were used separately to train the GMM model and extract the FV features. The GMM model was fit utilizing the training set. Fisher vectors were *optimized* employing Power Normalization (PN) and L2-Normalization before training the data; in [33] the authors obtained good performances using this feature pre-processing technique.

The x-vector network was fitted using the training data and its augmented version following the methodology employed in [13]; likewise, we used the same network topology proposed there. Basically, from the original training data, two augmented versions were added, i.e. noise and reverberation. From additive noises and reverberation, two of the following types of augmentation were chosen arbitrarily: babble, music, noise, and reverberation. The first three types correspond to simply adding or fitting a kind of noise to the original utterances, while the fourth one involves a convolution of room impulse responses with the audio, i.e. reverberation (see [13] for more details about the augmentation strategies used). From the artificially generated data, we chose a subset of 40000 utterances to train the DNN, which is roughly four times the number of original training samples. From the *segment6* layer of the DNN, we extracted 512-dimensional neural network embeddings (x-vectors) for the train, development, and test sets, respectively. As Snyder et al. [13], we also found that embeddings from *segment6* gave a better performance than those from *segment7* in our experiments.

Following the techniques suggested in [21], the parameter C of the SVM was set in the range: $10^{-5}, \ldots, 10^1$. Since the training and development sets are meant to be combined and used to train the final SVM model, we fused the above-mentioned sets and employed a Stratified k-fold Cross-Validation. We set $k = 10$ to find the best C. The training set has 5353 utterances labeled as *no-mask* and 5542 labeled as *mask*; the development set has 6666 and 7981, as *no-mask* and as *mask*, respectively. Namely, there is a slight class imbalance when combining both sets. As a result, there were 1504 more utterances labeled as *mask* in the combined set. Hence, we set the *class-weight* parameter of the SVM to *balanced*. In this way, the classifier adjusted the weights of the classes automatically. Before

Table 2. Experiment results. Scores are presented for x-vectors and FVs; both using MFCCs and PLPs. FoB stands for 'fusion of best' (fusion of the ComParE best configurations) [7]. The GMM size corresponds to the K value used for FV; for x-vectors this is not applicable. The dashes (-) in the UAR column indicate that the scores for those configurations are not available due to the limited number of trials for submissions defined by the organizers of the Challenge.

Feature	GMM size	UAR (%)		
		Dev	CV	Test
ComParE Baseline	-	-	-	71.81
x-vecs (MFCC)	-	56.86	65.21	-
x-vecs (MFCC-hires)	-	59.87	72.14	-
x-vecs (PLP)	-	58.46	64.80	-
FV (MFCC)	512	57.43	78.18	-
FV (PLP)	256	59.18	71.09	-
FV + FoB	512	-	-	70.30
FV + x-vecs (hires)	512	-	-	70.81
FV + x-vecs (hires) + FoB	512	-	-	74.92

classification, all the features were standardized by removing their means and scaling to unit variance.

In addition, we carried out a *late fusion* of our best configurations. Moreover, we also fused our best configurations with those posteriors from *'fusion of best'* of the sub-challenge [7].

5   Results and Discussion

As Table 2 shows, the FV representations produced better performances in the evaluation (i.e. Dev and CV) phase than the x-vectors embeddings did. However, this is mainly true for the CV scores, where FV achieved UAR scores above 70%. Overall, the configuration *FV (MFCC)* attained the best CV score (78.18%). On the other hand, the best configuration for the x-vectors embeddings was that of *high resolution* MFCCs (i.e. MFCC-hires), which gave a 72.14%. In contrast, when we evaluated the features using just the development set, x-vectors presented better scores; nevertheless, the difference compared to those of FV was not significant. Although in this study we did not rely on the development scores to find the best C value for the SVM, we still report the scores obtained when evaluating on this dataset (see Table 2). It should be added that we chose the best C based on the Stratified 10-fold CV experiments.

Furthermore, the FV encodings yielded a significant performance improvement when applying PN and L2-normalization before fitting them (see also [33]). However, here, just the best configurations are reported (the improved FVs). PN reduced the effect of the features that become more sparse as the value of K

increased. Also, L2-normalization helped to alleviate the problem of having different utterances with (relatively) distinct amounts of background information projected into the extracted features. This mainly enhanced the prediction performances. Also, it was found that the higher the number of K, the higher the UAR score. This means that these two are directly proportional to each other. In our study, both MFCC and PLP achieved their best configurations when using a large value for K (512 and 256, respectively). Likewise, MFCC-hires gave a better frame-level feature quality (for the x-vectors) than the standard MFCC configuration. This can be attributed to the DNN training phase, where the neural network exploits in a better way the larger and less correlated frame-level representations.

Table 2 lists the final scores. The fusion of the posteriors of FV_{512} with those of the *fusion of best* (from the challenge) attained a UAR score of 70.3% on the test set. Likewise, the fusion of FV_{512} with x-vectors *(x-vecs$_{hires}$)* yielded a score of 70.8%. Finally, the fusion of FV_{512} with x-$vecs_{hires}$ along with *FoB* provided a UAR score of 74.9% on the test set.

6   Conclusions

Here, we studied the performance of x-vector and Fisher vector representations as a contribution to the *Mask* Sub-Challenge of the INTERSPEECH 2020 ComParE. These representations were extracted from two different types of frame-level features: MFCC and PLP. As for the FV encodings, we found that MFCCs presented a superior type of frame-level traits of the recordings than the PLP did. Regarding the x-vectors, the configuration of *MFCC-hires* was found to be better than those of the standard MFCC and PLP. Also, we found that PN and L2-Normalization enhanced the quality of the FVs. Although the FV gave better quality features than x-vectors for this particular dataset, x-vectors also captured meaningful phonatory with articulatory information, as their scores are competitive. Moreover, we found that the fusion of our best configurations increased the performance of the final predictions. To conclude, our workflow outperformed the official baseline scores of the *Mask* Sub-Challenge [7]; besides, our feature extraction approach appears to be simpler than those from [7].

References

1. J. V. Egas-López, J. R. Orozco-Arroyave, and G. Gosztolya, "Assessing Parkinson's Disease From Speech Using Fisher Vectors," *Proceedings of Interspeech*, 2019.
2. G. Gosztolya, A. Bagi, S. Szalóki, I. Szendi, and I. Hoffmann, "Identifying schizophrenia based on temporal parameters in spontaneous speech," in *Proceedings of Interspeech*, Hyderabad, India, Sep 2018, pp. 3408–3412.
3. T. Grósz, R. Busa-Fekete, G. Gosztolya, and L. Tóth, "Assessing the degree of nativeness and Parkinson's condition using Gaussian Processes and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Dresden, Germany, Sep 2015, pp. 1339–1343.

 4. G. Gosztolya, T. Grósz, and L. Tóth, "General utterance-level feature extraction for classifying crying sounds, atypical & self-assessed affect and heart beats," in *Proceedings of Interspeech*, Hyderabad, India, Sep 2018, pp. 531–535.

 5. G. Gosztolya, T. Grósz, G. Szaszák, and L. Tóth, "Estimating the sincerity of apologies in speech by DNN rank learning and prosodic analysis," in *Proceedings of Interspeech*, San Francisco, CA, USA, Sep 2016, pp. 2026–2030.

 6. B. Schuller, A. Batliner, S. Steidl, and D. Seppi, "Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge," *Speech Communication*, vol. 53, no. 9-10, pp. 1062–1087, 2011.

 7. B. W. Schuller, A. Batliner, C. Bergler, E.-M. Messner, A. Hamilton, S. Amiriparian, A. Baird, G. Rizos, M. Schmitt, L. Stappen, H. Baumeister, A. D. MacIntyre, and S. Hantke, "The INTERSPEECH 2020 Computational Paralinguistics Challenge: Elderly emotion, Breathing & Masks," in *Proceedings of Interspeech*, Shanghai, China, September 2020, p. 5 pages, to appear.

 8. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

 9. N. S. Ibrahim and D. A. Ramli, "I-vector extraction for speaker recognition based on dimensionality reduction," *Procedia Computer Science*, vol. 126, pp. 1534–1540, 2018.

10. D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Twelfth annual conference of the international speech communication association*, 2011.

11. J. V. Egas-López, L. Tóth, I. Hoffmann, J. Kálmán, M. Pákáski, and G. Gosztolya, "Assessing Alzheimer's Disease from Speech Using the i-vector Approach," in *Proceedings of SPECOM*.   Springer, 2019.

12. N. Cummins, J. Epps, V. Sethu, and J. Krajewski, "Variability compensation in small data: Oversampled extraction of i-vectors for the classification of depressed speech," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 970–974.

13. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker verification," in *Proceedings of ICASSP*, 2018, pp. 5329–5333.

14. A. Silnova, N. Brummer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast variational bayes for heavy-tailed PLDA applied to i-vectors and x-vectors," *arXiv preprint:1803.09153*, 2018.

15. D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-speaker conversations using x-vectors," in *Proceedings of ICASSP*, 2019.

16. J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," *arXiv preprint arXiv:1806.05622*, 2018.

17. O. Novotnỳ, O. Plchot, P. Matejka, L. Mosner, and O. Glembek, "On the use of x-vectors for Robust Speaker Recognition." in *Proceedings of Odyssey*, 2018, pp. 168–175.

18. S. Zargarbashi and B. Babaali, "A multi-modal feature embedding approach to diagnose Alzheimer's disease from spoken language," *arXiv preprint arXiv:1910.00330*, 2019.

19. R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-vectors meet emotions: A study on dependencies between emotion and speaker verification," in *Proceedings of ICASSP*, 2020, pp. 7169–7173.

20. D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," *arXiv preprint arXiv:1909.06351*, 2019.
21. T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proceedings of NIPS*, Denver, CO, USA, 1998, pp. 487–493.
22. Y. Song, J. J. Zou, H. Chang, and W. Cai, "Adapting Fisher vectors for histopathology image classification," in *Proceedings of ISBI*.   IEEE, 2017, pp. 600–603.
23. F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *Proceedings of SPECOM*.   Springer, 2018, pp. 198–208.
24. D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network embeddings for text-independent speaker verification," in *Proceedings of Interspeech*, 2017.
25. D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep Neural Network-based speaker embeddings for end-to-end speaker verification," in *Proceedings of SLT*, 2016, pp. 165–170.
26. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The Kaldi speech recognition toolkit," *Proceedings of ASRU*, 01 2011.
27. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
28. X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of Visual Words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.
29. P. J. Moreno and R. Rifkin, "Using the Fisher kernel method for web audio classification," in *Proceedings of ICASSP*, Dallas, TX, USA, 2010, pp. 2417–2420.
30. Y. Tian, L. He, Z. yi Li, W. lan Wu, W.-Q. Zhang, and J. Liu, "Speaker Verification using Fisher vector," in *Proceedings of ISCSLP*, Singapore, Singapore, 2014, pp. 419–422.
31. Z. Zajíc and M. Hrúz, "Fisher Vectors in PLDA Speaker Verification system," in *Proceedings of ICSP*, Chengdu, China, 2016, pp. 1338–1341.
32. H. Kaya, A. A. Karpov, and A. A. Salah, "Fisher Vectors with cascaded normalization for paralinguistic analysis," in *Proceedings of Interspeech*, 2015, pp. 909–913.
33. J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher Vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, pp. 222–245, 2013.
34. F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of CVPR*, 2007.
35. D. C. Smith and K. A. Kornelson, "A comparison of Fisher vectors and Gaussian supervectors for document versus non-document image classification," in *Applications of Digital Image Processing XXXVI*, vol. 8856.   International Society for Optics and Photonics, 2013, p. 88560N.
36. C.-C. Chang and C.-J. Lin, "LIBSVM: A library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.
37. A. Vedaldi and B. Fulkerson, "VLFeat: an open and portable library of Computer Vision algorithms," in *Proceedings of ACM Multimedia*, 2010, pp. 1469–1472.