Predicting a Cold from Speech Using Fisher Vectors; SVM and XGBoost as Classifiers

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

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

Abstract. Screening a *cold* may be beneficial in the sense of avoiding the propagation of it. In this study, we present a technique for classifying subjects having a cold by using their speech. In order to achieve this goal, we make use of frame-level representations of the recordings of the subjects. Such representations are exploited by a generative Gaussian Mixture Model (GMM) which consequently produces a fixed-length encoding, i.e. Fisher vectors, based on the Fisher Vector (FV) approach. Afterward, we compare the classification performance of the two algorithms: a linear kernel SVM and a XGBoost Classifier. Due to the data sets having a high class imbalance, we undersample the majority class. Applying Power Normalization (PN) and Principal Component Analysis (PCA) on the FV features proved effective at improving the classification score: SVM achieved a final score of 67.81% of Unweighted Average Recall (UAR) on the test set. However, XGBoost gave better results on the test set by just using *raw* Fisher vectors; and with this combination we achieved a UAR score of 70.43%. The latter classification approach outperformed the original (non-fused) baseline score given in 'The INTERSPEECH 2017 Computational Paralinguistics Challenge'.

Keywords: Fisher vectors · Speech processing · SVM · XGBoost · Cold assessment · Computational paralinguistics

1 Introduction

Identifying cold or other related illnesses with similar symptoms may be beneficial when assessing them; as it could be a way of avoiding the spread of a specific kind of viral infection of the nose and throat (upper respiratory tract). Upper respiratory tract infection (URTI) affects the components of the upper airway. URTI can be thought as of a common cold, a sinus infection, among others. Screening a cold directly from the speech of subjects can create the possibility of monitoring (even from call-centers or telephone communications), and predicting their propagation. In contrast with Automatic Speech Recognition (ASR), which focuses on the actual *content* of the speech of an audio signal, computational paralinguistics may provide the necessary tools for determining the *way*

A. Karpov and R. Potapova (Eds.): SPECOM 2020, LNAI 12335, pp. 145–155, 2020.
https://doi.org/10.1007/978-3-030-60276-5_15

the speech is spoken. Various studies have offered promising results in this field: diagnosing neuro-degenerative diseases using the speech of the patients [6,7,10]; the classification of crying sounds and heart beats [13]; or even the estimation of the sincerity of apologies [12]. Hence, we focus on finding certain patterns hidden within the speech of the *cold* recordings and not on what the speakers actually said.

Here, we make use of the Upper Respiratory Tract Infection Corpus (URTIC) [26] to classify speakers having a cold. Previous studies applied various approaches for classifying *cold* subjects on the same corpus; for example, Gosztolya et al. employ Deep Neural Networks for feature extraction for such purpose [11]. Huckvale and Beke utilizated voice features for studying changes in health [14]; furthermore, Kaya et al. [16] introduced the application of a weighting scheme on instances of the corpus, employing Weighted Kernel Extreme Learning Machine in order to handle the imbalanced data that comprises the URTIC corpus.

In this study, frame-level features (Mel-frequency cepstral coefficients), extracted from the utterances, are utilized to fit a generative Gaussian Mixture Model (GMM). Next, the computation of low-level patch descriptors together with their deviations from the GMM give us an encoding (features) called the Fisher Vector. FV features are learned using SVM and XGBoost as binary classifiers, where the prediction is *cold* or *healthy*. In order to search for the best parameters of both SVM and XGBoost, Stratified Group k-fold Cross Validation (CV) was applied on the training and development sets. Unweighted Average Recall (UAR) scoring was used to measure the performance of the model. To the best of our knowledge, this is the first study that uses a FV representation to detect a cold from human speech.

In the next part of our study we also show that PN and L2-normalization over the Fisher vectors have a beneficial effect on the SVM classification scores. PN reduces the *sparsity* of the features; L2-normalization is a valid technique that can be applied to any high-dimensional vector; and moreover, it improves the prediction performance [25]. Likewise, PCA also affects positively to the performance of Support Vector Machines (SVM) due to its effects of feature decorrelation as well as dimension reduction.

The combination of all three feature pre-processing methods gave the best scoring with respect to the SVM classifier. However, XGBoost did not produce competitive scores when any kind of feature pre-processing was employed before training the model. Namely, employing the same feature-treatments (PCA, PN, L2-normalization) as SVM to the XGBoost classifier led to a decrease in performance. Mentioned algorithm showed better results when learning from *raw* features. Thus, there was no need for any feature processing prior to training, owing to the fact that decision tree algorithms do not necessitate so. We show that our system produces better UAR scores relative to the baseline individual methods reported in the 'The INTERSPEECH 2017 Computational Paralinguistics Challenge' [26] for the Cold sub-challenge.

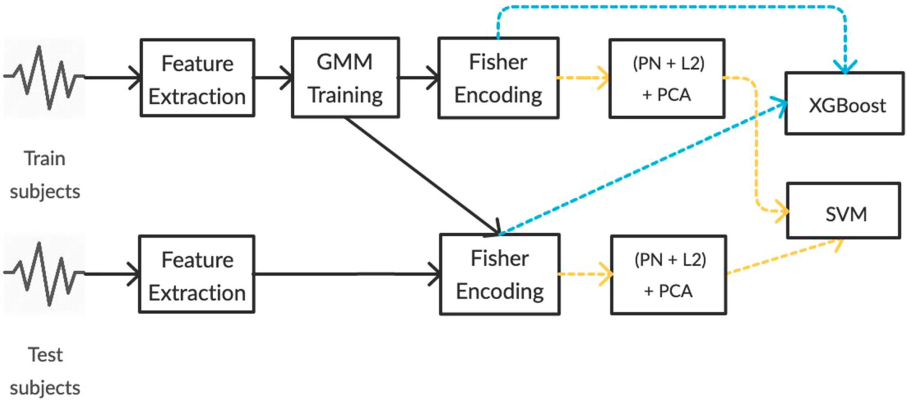

Fig. 1. The generic methodology applied in our work.

2 Data

The Upper Respiratory Tract Infection Corpus (URTIC) comprises recordings of 630 speakers: 382 male, 248 female, and a sampling rate of 16kHz. Recordings were held in quiet rooms with a microphone/headset/hardware setup. The tasks performed by the speakers were as follows: reading short stories, for example, *The North Wind and the Sun* which is well known in the phonetics area; producing voice commands such as numbers or driver assistant controlling commands; and narrating spontaneous speech. The number of tasks varied for each speaker. Although the sessions lasted up to 2 hours, the recordings were split into 28652 chunks of length 3 to 10 seconds. The division was done in a speaker-independent fashion, so each set had 210 speakers. The training and development sets were both comprised of 37 subjects having a cold and 173 subjects not having a cold [26]. The train, development, and test datasets are composed of 9505, 9596, and 9551 recordings respectively.

3 Methods

As outlined in Fig. 1, our workflow is as follows:

1. MFCCs features are extracted from all the recordings.
2. The GMM is trained using the MFCCs belonging to the training dataset.
3. The FV encoding (Fisher vectors) is performed for all the MFCCs of each utterance.
4. Classification:
   (a) Fisher vectors are processed using Power Normalization and L2 normalization; Support Vector Machines carried out the classification process using the new scaled features.
   (b) *Raw* Fisher vectors are fed to XGBoost for classification.

3.1 Feature Extraction

The frame-level features we utilized were the well-known MFCCs. We used a dimension of 20, plus their first and second derivatives, with a frame length of 25 ms and a frame shift of 10 ms.

3.2 The Fisher Vector (FV) Approach

This procedure can be viewed as an image representation that pools local image descriptors, e.g. Scale Invariant Feature Transform (SIFT). A SIFT feature is a selected image region with an associated descriptor (a descriptor can be thought as of a histogram of the image). In contrast with the Bag-of-Visual-Words (BoV, [22]) technique, it assigns a local descriptor to elements in a visual dictionary, obtained utilizing a Gaussian Mixture Model. 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. Applied to this study, such approach becomes quite practical because the length of the speech utterances are subject to vary.

The FV approach has been widely used in image representation and it can achieve high performance [15]. In contrast, just a handful of studies use FV in speech processing, e.g. for categorizing audio-signals as speech, music and others [20], for speaker verification [30,34], for determining the food type from eating sounds [17], and even for emotion detection [9]. These studies demonstrate the potential of achieve good classification performances in audio processing.

Fisher Kernel (FK). It 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 [15]:

$$G_\lambda^X = \bigtriangledown_\lambda \log \upsilon_\lambda(X), \tag{1}$$

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 [25]. Thus, such a gradient describes the way the parameter υ_λ should be changed in order to best fit the data X. A way to measure the similarity between two points X and Y by means of the FK can be expressed as follows [15]:

$$K_{FK}(X, Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y. \tag{2}$$

Since F_λ is positive semi-definite, $F_\lambda = F_\lambda^{-1}$. Equation (3) shows how the Cholesky decomposition $F_\lambda^{-1} = L_\lambda' L_\lambda$ can be utilized to rewrite the Eq. (2) in terms of the dot product:

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

where

$$G_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X). \tag{4}$$

Such a normalized gradient vector is the so-called *Fisher Vector* of X [25]. Both the FV G_λ^X and the gradient vector G_λ^X have the same dimension.

Fisher Vectors. Let $X = \{X_t, t = 1 \dots T\}$ be the set of D-dimensional local SIFT descriptors extracted from an image and let the assumption of independent samples hold, then Eq. (4) becomes:

$$G_\lambda^X = \sum_{t=1}^{T} L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X_t). \tag{5}$$

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:

$$X_t \rightarrow \varphi_{FK}(X_t) = L_\lambda \bigtriangledown_\lambda \log \upsilon_\lambda(X_t), \tag{6}$$

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

Hence, the FV approach extracts low-level local patch descriptors from the audio-signals' spectrogram. Then, with the use of a GMM with diagonal covariances we can model 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 [25]. 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 image 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 [23,25].

The FV approach can be compared with the traditional encoding method: BoV, and with a first order encoding method like VLAD Vector of Locally Aggregated Descriptors) [1]. In practice, BoV and VLAD are outperformed by FV due to its second order encoding property of storing additional statistics between codewords and local feature descriptors [28].

3.3 Classification with XGBoost and SVM

The classification of the data was carried out separately by two algorithms: XGBoost and SVM. In this section, we describe in a general manner these two approaches. SVM complexity parameter C was optimized by employing a Stratified Group k-fold Cross Validation using the train and development sets combined. For XGBoost parameters the same process was performed. Unweighted Average Recall (UAR) is the chosen metric due to the fact that it is more competent when having imbalanced datasets and also because it has been the de facto standard metric for these kinds of challenges [24,27].

As is widely known, a normal cross-validation gives the indices to split the data into train and test folds. In contrast, a stratified cross-validation applies the same principle but it preserves the percentage of samples for each class; and, a group k-fold cross-validation also has the same basis but it tries to keep the balance of different groups across the folds, so the same group will not be present in two distinct folds. Here, utterances from one speaker are treated as one group. The combination of these two different cross-validation approaches meant we could avoid having the same speaker in more than one specific fold while keeping the number of samples of each target class within that fold even.

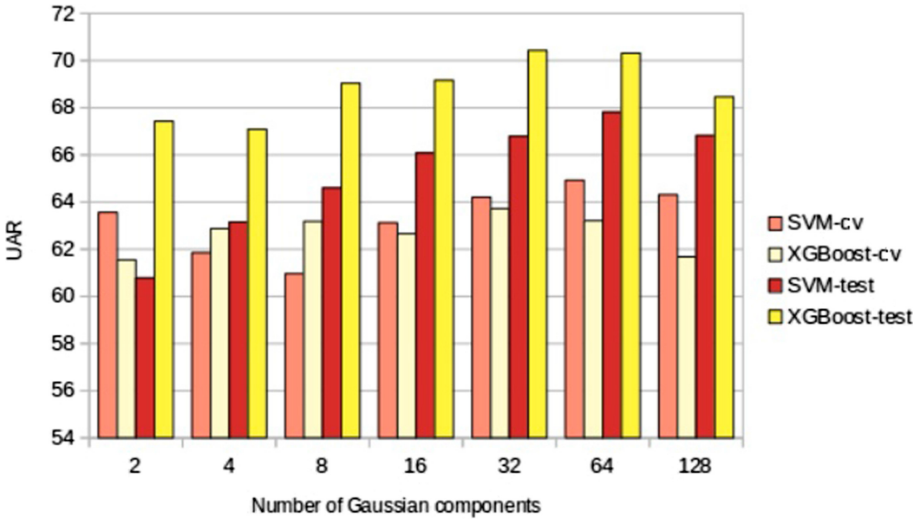

Fig. 2. UAR CV and test scores as a function of G_c for SVM and XGBoost using FVs.

XGBoost. This library is an implementation based on Gradient Boosting Machines (GBM) [8]. GBM is a regression/classification algorithm which makes use of an ensemble of *weak* models, i.e. small decision trees, to make predictions. A decision tree ensemble in XGBoost is a set of CARTs (Classification and Regression Trees). Put simply, GBM sequentially adds *decision tree* models to correct the predictions made by the previous models, and based on gradient descent, it minimizes the loss function. This is continued until the objective function (training loss and regularization) finds that no further improvement can be done [21]. Both XGBoost and GBM, basically act in the same manner; however, the main difference between these two is that XGBoost, in order to control over-fitting, employs a more regularized model than GBM does.

This algorithm is widely used in machine learning mostly due to its scaling capability and model performance; it was designed to exploit the limits of the computational resources for GBM algorithms [5]. Our decision to use XGBoost

was also influenced by its advanced capability for performing model tuning. We can see the performance of XGBoost in [19,32,33], where the authors report high scores using such algorithm when applied to speech-related classification tasks. In this study, we employed the Python implementation of XGBoost [5].

Support Vector Machines We relied on the libSVM implementation [3]. SVM was found to be robust even with a large number of dimensions and it was shown to be efficient when fitting them on FV [25,29]. To avoid overfitting due to having a large number of meta-parameters, we applied a linear kernel.

4 Experiments and Results

The GMM used to compute the FVs operated with a different number of components, G_c ranged from $2, 4, 8$ to 128. Here, the construction of the FV encoding was performed using a Python-wrapped version of the VLFeat library [31]. The dataset suffers from high class-imbalance, which could affect the performance of either of the classifiers. The training dataset comprises 9505 recordings: 8535 (89.8%) as *healthy* and the rest, 970 (10.2%) as *cold*. We relied on a random undersampling technique that reduces the number of samples associated with all classes, to the number of samples of the minority class, i.e. *cold*. We employed imbalanced-learn [18], a Python-based tool which offers several resampling methods for between-class imbalance. For the SVM, the complexity value (C) was set in the range $10^{\{-5,-4,\ldots,0,1\}}$.

As a baseline, we utilized the ComParE functionals that were originally presented and described in [26]. As Table 1 shows, these representations achieved an UAR score of 69.30% on the test set, which is slightly higher than the score achieved with FV representations (67.81%). The SVM classifier gave better results using FVs with Power Normalization (PN) and L2-Normalization, along with PCA: UAR score of 67.81% (see Table 1). PCA was applied using the 95% of the variance, which apart from decorrelating the FV features, also helped with both the computation (lower memory consumption) and the discrimination task. This method is also described in [4]. We saw that PN helped to reduce the impact of the features that become more sparse as the number of Gaussian components increases. Meanwhile, L2-normalization helped to alleviate the effect of having different utterances with distinct amounts of background information projected into the extracted features, which attempts to improve the prediction performance. Also, we employed a late fusion of the posterior probabilities. We combined the ComParE functionals SVM-posteriors with those that gave the highest scores when using SVM on FVs; the result was a better UAR score: 70.71%.

For XGBoost, we just utilized the non-preprocessed Fisher vector features and performed a grid search to find the best parameters. To control overfitting, we tuned the parameters that influence the model complexity: the gamma value, which represents the minimum loss required to split further on a leaf; the maximum depth for each tree (the higher the value, the higher the complexity); and the minimum child weight, that is, the minimum sum of weights needed in a

child. Also, the learning rate and the number of estimators (number of trees) were tuned, these two having an inverse relation: the higher the learning rate the smaller the number of trees that have to be defined, and vice-versa.

As shown in Fig. 2, the classifiers discriminate better the data as the value of G_c increases. However, the highest G_c did not give the best UAR score. SVM classified better when the Fisher vectors were encoded using 64 Gaussian components, while a smaller number of G_c was needed for XGBoost (32). Stratified k-fold CV (on the combined train and development data) with $k = 10$ was applied for the hyper-parameter tuning of both algorithms. Due to XGBoost basically being an ensemble of regression trees, its posterior probability values are not really meaningful, hence we did not perform any kind of fusion with them. In spite of this, such algorithm achieved a score of 69.59% with the ComParE feature set and 70.43% using the Fisher vector features; the former outperformed the non-fused highest score of SVM (67.81%) and it is slightly lower than the fused one (70.17%), while the latter surpassed both of them. Furthermore, these scores surpassed the non-fused baseline and are around the fused baseline score given in [26] (see Table 1).

Table 1. UAR scores obtained using XGBoost and SVM on the URTIC Corpus.

2* Features	2* GMM size	Performance (%)	
		CV	Test
		SVM	
ComParE	–	64.20%	69.30%
Fisher vectors	64	63.98%	66.12%
Fisher vectors + PCA	64	64.72%	67.65%
Fisher vectors (+PN+L2) + PCA	64	64.92%	67.81%
Fusion: ComParE + FV(+PN+L2+PCA)	–/64	63.01%	70.17%
		XGBoost	
ComParE	–	62.19%	69.59%
Fisher vectors	32	63.71%	70.43%

5 Conclusions and Future Work

Here, we showed how well the Fisher vector encoding allows frame-level features to classify speaking subjects with a cold. We utilized two different classification algorithms (SVM and XGBoost) that used FV as input features. We showed that such features trained on both algorithms outperform original baseline scores given in [26] and they are highly competitive with those reported in [2,14]. Moreover, our approach offers a much simpler pipeline than the above-mentioned studies. We found that we got a better SVM performance when we applied

feature pre-processing before starting the train/classification phases. Namely, it was shown that both L2-Normalization and Power Normalization produced an increased prediction performance. Also, PCA played a relevant role in decorrelating the features and increasing the model's performance. We demonstrated the usefulness of the fusion of SVM posterior probabilities which yielded even better UAR results. In contrast, XGBoost did not need any pre-processing or any kind of fusion to achieve and surpass SVM scores in our study. Yet, one disadvantage of XGBoost was the significant number of parameters that have to be tuned. This can slow down the parameter-tuning phase especially if there is no GPU available. In our next study, we plan to apply the methodology presented here on different kinds of paralinguistic corpora.

Acknowledgments. This study was partially funded by the National Research, Development and Innovation Office of Hungary via contract NKFIH FK-124413 and by the Ministry for Innovation and Technology, Hungary (grant TUDFO/47138-1/2019-ITM). G. Gosztolya was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the Hungarian Ministry of Innovation and Technology New National Excellence Program ÚNKP-20-5.

References

1. Arandjelovic, R., Zisserman, A.: All about VLAD. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1578–1585 (2013)
2. Cai, D., Ni, Z., Liu, W., Cai, W., Li, G., Li, M.: End-to-end deep learning framework for speech paralinguistics detection based on perception aware spectrum. In: Proceedings of Interspeech, pp. 3452–3456 (2017)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 1–27 (2011)
4. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: British Machine Vision Conference, vol. 2, pp. 76.1–76.12, November 2011
5. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining abs/1603.02754, pp. 785–794 (2016)
6. Egas-López, J.V., Orozco-Arroyave, J.R., Gosztolya, G.: Assessing Parkinson's disease from speech using fisher vectors. In: Proceedings of Interspeech (2019)
7. Egas López, J.V., Tóth, L., Hoffmann, I., Kálmán, J., Pákáski, M., Gosztolya, G.: Assessing Alzheimer's disease from speech using the i-vector approach. In: Salah, A.A., Karpov, A., Potapova, R. (eds.) SPECOM 2019. LNCS (LNAI), vol. 11658, pp. 289–298. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26061-3_30
8. Friedman, J.H.: Greedy function approximation: a Gradient Boosting Machine. Ann. Stat. **29**, 1189–1232 (2001)
9. Gosztolya, G.: Using the Fisher vector representation for audio-based emotion recognition. Acta Polytechnica Hungarica **17**(6), 7–23 (2020)
10. Gosztolya, G., Bagi, A., Szalóki, S., Szendi, I., Hoffmann, I.: Identifying schizophrenia based on temporal parameters in spontaneous speech. In: Proceedings of Interspeech, Hyderabad, India, pp. 3408–3412, September 2018

11. Gosztolya, G., Busa-Fekete, R., Grósz, T., Tóth, L.: DNN-based feature extraction and classifier combination for child-directed speech, cold and snoring identification. In: Proceedings of Interspeech, Stockholm, Sweden, pp. 3522–3526, August 2017
12. Gosztolya, G., Grósz, T., Szaszák, G., Tóth, L.: Estimating the sincerity of apologies in speech by DNN rank learning and prosodic analysis. In: Proceedings of Interspeech, San Francisco, CA, USA, pp. 2026–2030, September 2016
13. Gosztolya, G., Grósz, T., Tóth, L.: General utterance-level feature extraction for classifying crying sounds, atypical and self-assessed affect and heart beats. In: Proceedings of Interspeech, Hyderabad, India, pp. 531–535, September 2018
14. Huckvale, M., Beke, A.: It sounds like you have a cold! testing voice features for the interspeech 2017 computational paralinguistics cold challenge. In: Proceedings of Interspeech, International Speech Communication Association (ISCA) (2017)
15. Jaakkola, T.S., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Proceedings of NIPS, Denver, CO, USA, pp. 487–493 (1998)
16. Kaya, H., Karpov, A.A.: Introducing weighted kernel classifiers for handling imbalanced paralinguistic corpora: snoring, addressee and cold. In: Interspeech, pp. 3527–3531 (2017)
17. Kaya, H., Karpov, A.A., Salah, A.A.: Fisher vectors with cascaded normalization for paralinguistic analysis. In: Proceedings of Interspeech, pp. 909–913 (2015)
18. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(1), 559–563 (2017)
19. Long, J.M., Yan, Z.F., Shen, Y.L., Liu, W.J., Wei, Q.Y.: Detection of Epilepsy using MFCC-based feature and XGBoost. In: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–4. IEEE (2018)
20. Moreno, P.J., Rifkin, R.: Using the Fisher kernel method for web audio classification. In: Proceedings of ICASSP, Dallas, TX, USA, pp. 2417–2420 (2010)
21. Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. Front. Neurorob. **7**, 21 (2013)
22. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput. Vis. Image Underst. **150**, 109–125 (2016)
23. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007.https://doi.org/10.1109/CVPR.2007.383266
24. Rosenberg, A.: Classifying skewed data: importance weighting to optimize average recall. In: Proceedings of Interspeech, pp. 2242–2245 (2012)
25. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vision **105**(3), 222–245 (2013). https://doi.org/10.1007/s11263-013-0636-x
26. Schuller, B., et al.: The Interspeech 2017 computational paralinguistics challenge: addressee, cold and snoring. In: Computational Paralinguistics Challenge (ComParE), Interspeech 2017, pp. 3442–3446 (2017)
27. Schuller, B.W., Batliner, A.M.: Emotion, Affect and Personality in Speech and Language Processing. Wiley, Hoboken (1988)
28. Seeland, M., Rzanny, M., Alaqraa, N., Wäldchen, J., Mäder, P.: Plant species classification using flower images: a comparative study of local feature representations. PLOS ONE **12**(2), 1–29 (2017)

29. Smith, D.C., Kornelson, K.A.: A comparison of Fisher vectors and Gaussian supervectors for document versus non-document image classification. In: Applications of Digital Image Processing XXXVI, vol. 8856, p. 88560N. International Society for Optics and Photonics (2013)
30. Tian, Y., He, L., Li, Z.Y., Wu, W.L., Zhang, W.Q., Liu, J.: Speaker verification using Fisher vector. In: Proceedings of ISCSLP, Singapore, pp. 419–422 (2014)
31. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 1469–1472. ACM (2010)
32. Wang, C., Deng, C., Wang, S.: Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. arXiv preprint arXiv:1908.01672 (2019)
33. Wang, S.-H., Li, H.-T., Chang, E.-J., Wu, A.-Y.A.: Entropy-assisted emotion recognition of valence and arousal using XGBoost classifier. In: Iliadis, L., Maglogiannis, I., Plagianakos, V. (eds.) AIAI 2018. IAICT, vol. 519, pp. 249–260. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92007-8_22
34. Zajíc, Z., Hrúz, M.: Fisher Vectors in PLDA speaker verification system. In: Proceedings of ICSP, Chengdu, China, pp. 1338–1341 (2016)