Assessing speaker-independent character information for acted voices

Mathias Quillot, Richard Dufour, Jean-François Bonastre

Assessing speaker-independent character information for acted voices

Mathias Quillot[0000−0002−5858−2416], Richard Dufour[0000−0003−1203−9108], and Jean-François Bonastre[0000−0001−7741−3346]

Université d'Avignon, Avignon 84140, France
{firstname}.{lastname}@univ-avignon.fr

Abstract. While the natural voice is spontaneously generated by people, the acted voice is a controlled vocal interpretation, produced by professional actors and aimed at creating a desired effect on the listener. In this work, we pay attention to the aspects of the voice related to the character played. We particularly focus on actors playing the same video game role in different languages. This article is based on a recent work which proposes to build a neural-network-based voice representation dedicated to the character aspects, namely *p*-vector. This representation is learnt from recordings only labeled with the acted character. It showed its ability to associate two vocal examples related to the same character, even if the character is unknown during the training phase. However, there is still a possible confusion between speaker and character dimension. To tackle this problem, We propose a protocol to highlight the speaker-independent part of the character information (SICI). We compare the original voice representation with an alternative where the information relating to the characters is neutralised. This experiment shows that performance is not a sufficient metric to assess the quality of a character representation. It also offers the first evidence of the SICI in the voice.

Keywords: voice casting · speaker recognition · character information · speaker-independent character information · speaker information.

1 Introduction

For decades, the voice has attracted considerable attention from researchers. In speech processing, several areas emerge, such as spoken language recognition [13], automatic speech recognition [14], speaker verification [3], emotion recognition [22], speech understanding [11], voice transformation [21] or conversion [5]. Research efforts in this quite diverse list of areas share one common trait, in terms of the raw material being worked on: most focus on natural voice recordings — spontaneous or read speech, telephone recordings, or speech resulting from human-machine dialogues (through, for example, voice assistants). In comparison, acted voice is poorly represented in speech processing, except in the paralinguistics [6, 20] or speech synthesis [10, 19] fields, where recordings

pronounced by professional actors, playing a specific emotion for instance, are frequently encountered.

Unlike natural speech, acted voice is a controlled vocal interpretation often encountered in audiovisual production. Directed by professional actors, it aims to create a desired effect for the viewer by making manifest the behavior of a fictional character or by facilitating their immersion. The voice is often distorted, sometimes overplayed, in order to make the desired expressive effect more audible. Aspects of the actor's interpretation not related to linguistic contents fall on the listener's side of a complex perception (cultural aspects and stereotypes are obviously there). This interpretation depends on the listeners themselves and their personal history but also on the specific context of each listening. This double complexity, in terms of production and perception, perhaps explains why the acted voice has such little presence in the literature on speech processing. The articles [7–9, 15–17] cited in this paper are the only ones dealing with the speech-processing-based automation of Voice Casting in the literature to our knowledge.

In this article, we wish to address some of this complexity and we start with the voices of professional actors playing characters for the gaming industry. We rely on [7], a recent work which defines the p-vector, a neural-network-based representation of the voice dedicated to the character aspects in acted voices. The application context in [7] (like in related previous works [8, 9]) is voice casting for audio dubbing. The final objective of this work is to assess how close an actor acting in a target language is to the voice of a given character, speaking in a source language. This can be referred to as character-based voice similarity. In contrast to [16, 17], which proposed a voice similarity system for the acted voice based on data labeled with speech classes (age, gender, emotions ...) by an expert, the p-vector approach does not use any expert labels. This representation is learnt from audio associated with the played character label, without any additional information.

Although these approaches deal with final works (video games), another one consists in working directly on the decision data from Artistic Directors. Recently, researchers from Warner Bros. evaluated their models on this kind of data. Unfortunately, these data are sensitive and their acquisition is not trivial since it requires to work on the critical voice casting process. This is why we decided to position ourselves in a task quite different from [15], since we do not use Artistic Director decisions.

In order to evaluate the efficiency of the character representation in the voice dubbing context, [8] proposes to detect whether a pair of speech extracts in the source and the target language belongs, or not, to the same character, knowing that the extracts within a pair are always spoken by different speakers. For that purpose, a Siamese-network-based binary classifier like in [1, 12] is trained on the top of p-vectors. Experimental results have shown the ability of the p-vector representation to associate two voice excerpts related to the same character, even when a particularly difficult scenario is employed, where the character and the two speakers of a given pair of recordings are completely unknown during the

training phase. However, there are several potential biases in this experiment, like the fact that a given character is represented by a unique pair of actors, the length of the speech extracts, their linguistic content or the influence of speaker-specific information [8]. Even if the proposed protocol allows us to overcome these biases as much as possible, the latter remains debatable because there is still a possible entanglement between the speaker dimension and the character aspects.

Hence, in order to tackle the bias issue of previous evaluation methods, this article proposes a totally new approach to assess the presence of speaker-independent character information (SICI) encoded by a character voice representation like p-vectors. This evaluation can help verifying if systems do not only learn to associate speaker identity but also if they base their decision on character-specific aspects. The main idea is to start with a solution close to the one proposed in [7, 8]. Then, we build an alternative model where information about characters is neutralized during the training by swapping the character labels of the voice recordings. This swapping is done at the actor level: all the recordings pronounced by an actor are now wrongly associated to a character label, chosen randomly. It is worth remembering that, because of the voice dubbing context of this work, this is tantamount to breaking the link between an original actor and the dubber associated to him. When comparing the performance of the original system against the second one, we expect to observe a loss proportional to the part of the neutralized (speaker-independent) character information.

The article is organized as follows. In Section 2, we give a brief overview of the p-vector representation and the solution to evaluate it. In Section 3, we detail the central part of this work, our protocol to estimate the relative quantity of character information. We also present results obtained by applying our modified data. For reproducibility purposes, scripts and models are available on GitHub[1]. Results are then discussed in Section 4. We finally conclude in Section 5.

2 Character representation extraction and evaluation

In this section, we present an overview of our character similarity system. The p-vector character-based voice representation, close to that proposed in [7, 8], is then described, as well as the voice-similarity based approach used to evaluate it. We also propose to evaluate these systems on a closed protocol where the test speakers (and therefore characters) are known to the training phase.

2.1 Character similarity system overview

Figure 1 gives an overview of the complete character voice similarity production chain used in this work. A p-vector (*character-oriented representation*) is

[1] https://github.com/LIAvignon/specom2021-assessing-speaker-independent-character-information-for-acted-voices

obtained from a representation of the speech signal (*sequence extractor*). Then, the decision module (*decision*) takes as input a pair of *p*-vectors and generates a *score* about the character similarity between them. Finally, this score is compared to a decision threshold in order to obtain a binary decision.

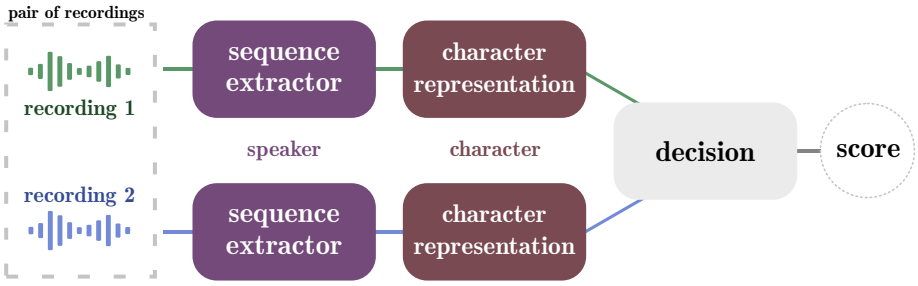

Fig. 1. Production chain of scoring character voice similarity system from the signal to similarity score.

2.2 Character-oriented representation

This section describes the character-oriented voice representation, named *p*-vector, introduced in [7]. These *p*-vectors are built from a representation of speech signal and are intended to highlight character information of a given voice recording.

Each recording $r \in train$ is associated with the acted character. We train a Multi-Layer Perceptron (MLP) as classification system in order to recognize the character in a closed space according to the given recording, whatever the language in which it is acted. We give as input to the character extraction system an *i*-vector representation [4] of 400 dimensions obtained from the recording's signal (i.e., indicated as 'sequence extractor' in Figure 1; details of the extraction are explained in article [8]). The system calculates us back, for each class, the probability that the recording belongs to the class (i.e., the character label). Once the network is trained, [7] propose to use the last layer as embedding, before softmax, as the *p*-vectors.

The neural network is composed of four *Dense* layers, all with 256 neurons, accompanied by a tangent hyberbolic activation function and a dropout of 0.25, except the fourth which is used as embedding, and has only 64 neurons and a dropout of 0.5. A last layer of softmax at the end of the neural network is added. The algorithm we use to train the network is *adadelta* with the cost function *categorical crossentropy*. To avoid overfitting, we apply an early stopping with a minimum delta to 0.1 and a patience of 10 epochs to training algorithm.

2.3 Voice similarity model

As said before, in order to evaluate our character representation, we build a character-based voice similarity on top of it. The task consists in deciding if a pair of recordings, one in English and the other in French, belongs to the same character or not. We compute, for each pair of voices X, the score $H_f(X)$ and compare it to a threshold set at the *a posteriori* EER (Equal Error Rate). The voice similarity module is based on the Siamese Neural Network presented in [8].

Performance evaluation and confidence intervals The voice similarity system (i.e., indicated as 'decision' in Figure 1) is used in this work to verify the effectiveness of the character-oriented voice representation module (i.e., indicated as 'character representation' in Figure 1). The performance is computed as the binary accuracy of the voice similarity system, which is the ratio between the number of correctly classified pairs over the total number of classified ones.

We also use a *test of proportion* to assess the statistical significance of accuracy differences. This method takes two proportions p_1 and p_2 and evaluates the hypothesis H_1 saying that the proportions are equivalents. A confidence interval is computed using p_2 and the hypothesis is confirmed if $\hat{p}$ is in this interval. Otherwise, the hypothesis is rejected. We compare accuracy a_1 and a_2 of two given systems by applying this test with a significance level of 5% and with $p_1 = a_1$ and $p_2 = a_2$.

2.4 Corpus description

The main corpus is composed of voice recordings coming from the *Mass-Effect 3* role-playing game. Originally released in English, the game has been translated and revoiced in several other languages. In our experiments, we use the English and French versions of the audio sequences, representing about 7.5 hours of speech in each language. Segments (or recordings) are 3.5 seconds long on average. A character is then defined by a unique French-English couple of two distinct speakers. To avoid any bias in terms of speaker identity, we consider only a small subset, where we are certain that none of the actors play more than one character. A single audio segment corresponds to a unique speaking slot from an actor in a particular language. We have then applied a filter that keeps only recordings for which the duration is greater than 1 second. Finally, we only keep 16 characters for which we have the largest number of recordings.

Contrary to the article [8] that proposes to apply a 4-fold protocol, we decided in this paper to keep the 16 characters, and their 32 corresponding speakers, for both training and testing phases. We split the corpora in three subsets: training (*train*), validation (*val*), and test (*test*) using a 80/10/10 *rule*. All these subsets are composed of different recordings but arising from the same 16 characters and 32 speakers. To build respectively the *train*, *val* and *test* subsets, we randomly select for each character 144, 18, and 18 recordings, while balancing the number of French and English recordings. We have then respectively a total of $2,304$, 288 and 288 recordings.

For each subset, we build pairs of recordings where the first element is a voice segment belonging to an actor in the source language (English), and the second is a recording pronounced by another actor, the dubber, in the target language (French). We associate the class *target* to pairs of voices coming from the same character, and *non-target* otherwise. Pairs are made with randomly selected segments while balancing *targets* and *non-targets*. This pairing process is denoted *original data*. We have, for pairs respectively built from *train*, *val* and *test*, $165,888$, $2,592$ and $2,592$ pairs.

2.5 Performance of the *p*-vector representation

Table 1 shows the performance of the character-oriented voice similarity system (*prot 2*) built upon the *p*-vectors. Performance of a voice comparison system (*prot 1*) based only on the *i*-vectors (in this representation, no specific information about the characters played is used) and of a random system are provided for comparison purposes. A large difference of 7 points in accuracy (87% for *p*-vectors versus 80% for *i*-vectors) is noticed. Looking at the confidence intervals, this difference is strongly significant. It confirms the results observed in [7] (using a different protocol) where it was found that the *p*-vector representation seems to embed specific information about the played characters.

Table 1. Performance of *i*-vectors (*i*-v) and *p*-vectors (*p*-v) on original data. Random is the theoretical performance of random system. 95% confidence interval limits indicated in brackets.

	sequence extraction	character layer (*p*-v)	performance
random	×	×	0.50 [0.48, 0.52]
prot 1	*i*-v	×	0.80 [0.78, 0.82]
prot 2	*i*-v	original	0.87 [0.86, 0.88]

3 Estimation of the amount of character information in the *p*-vector representation

Previous section empirically proves that the *p*-vector representation improves the performance on a character-based similarity task. Nonetheless, this does not ensure completely that the model captures character information, as the *p*-vector representation is initially based on a speaker representation that may still embody speaker-related information. In order to verify that the improvement observed while using *p*-vector does not come from the ability to associate

voices, we propose as a main contribution in this article to train our acted voice similarity systems with misled data where the character information is supposed being neutralized.

3.1 Random association protocol

The random association protocol we propose consists in training a neural network on intentionally misled data. As shown by Figure 2, for each English actor A_i, we associate a new French actor D_j different from the initial dubber one D_i, while keeping the constraint that a French actor is only paired with an English one. To avoid gender bias, we still choose a pair of actors sharing the same gender (*male* or *female*). As a character is represented by a unique pair of speakers, it corresponds to a random labeling of the files in terms of characters. So, the speakers are still associated by pair (one English and one French) but they no longer belong to the same character inside a given pair.

The performance using this new association compared with the system without p-vectors should show the "speaker" power of the p-vector. A performance difference versus the character-based file pairing should indicate the part of character information embed into the p-vectors.

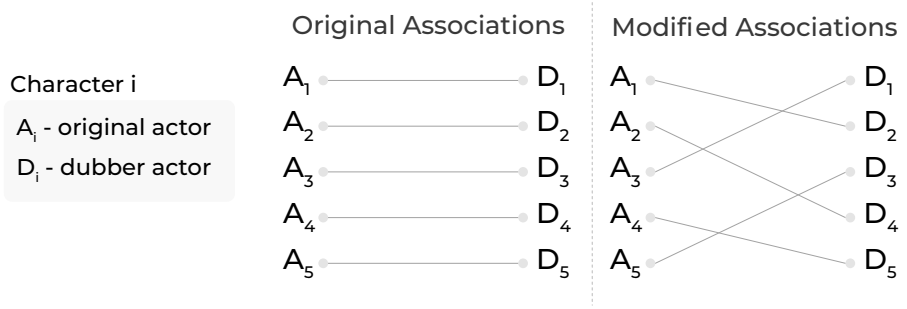

Fig. 2. Random association protocol where each actor A_i is originally associated to the dubber D_i.

3.2 Random associations subsets

As explained in Section 3.1, the random association protocol consists in intentionally switching the dubber associated with each original actor and then generating new subsets of voice pairs with the modified actor associations using the same steps as presented in Section 2.4. This random association dataset is noted *modified*.

Using this new modified protocol, two modules may be impacted, since they can be trained from the original or modified character labels: the p-vector representation (*character representation*) and the Siamese voice similarity system

(*decision*). We then train a *p*-vector extractor, denoted as *modified*, with the new randomly switched character labels for dubbing voices. In the same way, we also train a version of our voice-similarity Siamese system using the *modified* labels. Of course, when the voice similarity system is trained using the *modified* labels, the same *modified* speaker pairing is used to assess the performance (in the test dataset).

3.3 Experiments and results

Modifying protocol in order to highlight the Speaker-independent character information (SICI) Our first experiment consists in assessing the presence, or not, of SICI on voice representation. The modified protocol is a means of removing character information. Therefore, the absolute difference between the score of systems trained on original or modified data is a clue to assess the presence of SICI. With this in mind, we train an *i*-vector sequence extractor and then build a *p*-vector embedding on top of it by using original or modified data. The character representation is then evaluated with a Siamese neural network trained on character voice similarity as explained before. Table 2 summarizes the system performance. To validate that the accuracies of two systems are significantly different, the confidence intervals are written below the performance scores. In this table, *prot 3* and *4* respectively correspond to the modified version of *prot 1* and *2*. We also propose a mixed version named *prot 5* where *p*-vectors (character module) are trained using original data and are then evaluated on the modified dataset (decision module) in order to assess the presence of speaker information on the embedding.

Table 2. Performance (accuracy) of *i*-vector (*i*-v) and *p*-vector (*p*-v) representations on modified data. 95% confidence interval limits are given in brackets. The rows for "prot 1" and "prot 2" are repeated from Table 1 for the reader's convenience.

	sequence extractor	tying pairs		performance
		character layer (*p*-v)	decision layer	
prot 1	*i*-v	×	original	0.80 [0.78, 0.82]
prot 2	*i*-v	original	original	0.87 [0.86, 0.88]
prot 3	*i*-v	×	modified	0.80 [0.78, 0.82]
prot 4	*i*-v	modified	modified	0.84 [0.83, 0.85]
prot 5	*i*-v	original	modified	0.75 [0.73, 0.77]

Comparison with a neural network sequence extractor As we based our systems on *i*-vector speaker embedding, we also want to compare these results with a neural network sequence extractor. For this purpose, we build an *x*-vector extractor with the Kaldi [18] toolkit using the Voxceleb corpus [2]. We use it as a sequence extractor in place of *i*-vectors. Then, we train *p*-vectors and evaluate them with a Siamese neural network by following exactly the same original and modified protocols used for *i*-vectors. Table 3 presents the results: protocols from *prot 6* to *prot 10* respectively correspond to protocols from *prot 1* to *prot 5* where the only difference is the replacement of *i*-vectors by *x*-vectors approach as sequence extractor.

Table 3. Performance (accuracy) of *x*-vector (*x*-v) and *p*-vector (*p*-v) representations on modified data. 95% confidence interval limits are given in brackets.

	sequence extractor	tying pairs		performance
		character layer (*p*-v)	decision layer	
prot 6	*x*-v	×	original	0.85 [0.83, 0.87]
prot 7	*x*-v	original	original	0.90 [0.89, 0.91]
prot 8	*x*-v	×	modified	0.76 [0.74, 0.78]
prot 9	*x*-v	modified	modified	0.90 [0.89, 0.91]
prot 10	*x*-v	original	modified	0.77 [0.75, 0.79]

4 Discussion

The first part of our analysis will focus on the systems based on *i*-vectors, listed in Table 2. Since *prot 3* yields the same result as *prot 1* (0.80), we can conclude that neutralizing the character information has no effect on the accuracy of the systems. It confirms that the information encoded by *i*-vectors is mainly presented from a speaker angle, skewing the Siamese network. These latter consequently has difficulties finding speaker-independent character information (SICI).

We then analyze the contribution of the *p*-vectors on the information encoding. As we know, building *p*-vectors on top of *i*-vectors highlights the character information. This was demonstrated by the fact that *p*-vector system trained on original data both for the character and decision layers did outperform the one trained on *i*-vectors, respectively *prot 2* (0.87) and *prot 1* (0.80). We can also notice in the Table 2 that *p*-vectors bring SICI.

Table 2 also shows the accuracy obtained on *prot 5* (0.75). Even if *p*-vectors learn to associate speakers with the original associations, the Siamese neural

networks trained with modified data manage to find information that allows it to associate speakers. As a consequence, we assume that speaker information is present in p-vectors and that it is legitimate to wonder if this information does not skew the decision module.

Since neural network approaches are state of the art in literature about speaker representation, we have compared the use of i-vectors with a neural-based sequence extractor, the x-vectors. In Table 3, we can observe that all the scores are better than those obtained using i-vectors. While p-vectors built on top of the x-vectors, and the x-vectors themselves, seem to encode more speaker and character information, we can observe an inversion of behaviour when comparing with i-vector performance. Indeed, the absolute difference between *prot 6* (0.85) and *prot 8* (0.76) is about 9%, where similar systems for i-vectors did not display any differences. In addition, we observe no difference between *prot 7* and *prot 9* (0.90) while i-vectors performances show a significant absolute difference of about 3%.

As we observe that no systems trained on modified data perform better than those trained on original data, we assume that we are achieving the objective of neutralizing character information as expected. This consequently allows us to highlight the SICI to prevent speaker skewing.

These experiments also show that while systems based on x-vectors outperform those based on i-vectors (as shown in [7] and Tables 2 and 3), it does not necessarily mean that this encodes a better quality representation. Indeed, the peculiarities of our corpus facilitate the use of speaker information since each actor from each language only plays one character. The system can learn to associate together speaker identities and to disregard character information not related to the speaker dimension. This consequently makes character and speaker dimension really intertwined and difficult to disentangle. Thanks to our approach, new works will be able to assess more precisely their character-based model trained by speaker association and ensure that their system are not too much speaker-oriented.

5 Conclusion and future work

In this article, we proposed to highlight the speaker-independent part related to the character played in acted voices. We built up on p-vectors, a representation learning approach dedicated to character's information in acted voices. We used for evaluation purposes a Siamese network binary recognizer capable of deciding whether two voices are linked to the same character or not. We first went through previous paper experiments showing that p-vectors help to achieve this task. Next, we moved on to the first objective of this work, which was to assess whether p-vectors really capture information about the characters and do not just memorize the voices of the speakers. For this, we have designed a specific configuration capable of neutralizing information on the character in the p-vector while retaining intact its capacities for memorizing the speaker-related information.

Our experiment has shown that this configuration neutralizes the character and provides a good framework to analyze speaker bias from character-based systems. Thanks to this method, we have also shown that p-vectors can highlight part of character information related or not to speaker identity. However, we have also highlighted that performance is not a good indicator of representation quality. Indeed, the system achieving the best performances did not encode SICI, leading us to conclude that the system only learns to associate speaker identities.

In future works, we first want to extend our work to other audiovisual productions, such as movies, maybe less stereotypical than video-games. Second, the p-vector character-based representation may suffer from the representation of the speech sequence used in this work, the i-vectors. To overcome this limitation, we will work on end-to-end representations directly trained with the objective of focussing on character dimension.

6 Acknowledgements

This project is supported by the French National Research Agency (ANR) TheVoice grant (ANR-17-CE23-0025).

References

1. Chopra, S., Hadsell, R., LeCun, Y.: Learning a Similarity Metric Discriminatively, with Application to Face Verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 539–546. IEEE (2005)
2. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: Deep speaker recognition. In: Interspeech (2018)
3. Das, R.K., Prasanna, S.R.: Speaker Verification from Short Utterance Perspective: A Review. IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India) **35**(6), 599–617 (2018)
4. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech and Language Processing **19**(4), 788–798 (2011)
5. Ezzine, K., Frikha, M.: A comparative study of voice conversion techniques: A review. In: International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). pp. 1–6 (2017)
6. Gideon, J., Khorram, S., Aldeneh, Z., Dimitriadis, D., Provost, E.M.: Progressive neural networks for transfer learning in emotion recognition. In: Annual Conference of the International Speech Communication Association (INTERSPEECH). pp. 1098–1102 (2017)
7. Gresse, A., Quillot, M., Dufour, R., Bonastre, J.F.: Learning Voice Representation Using Knowledge Distillation For Automatic Voice Casting. In: Annual Conference of the International Speech Communication Association (INTERSPEECH) (2020)
8. Gresse, A., Quillot, M., Dufour, R., Labatut, V., Bonastre, J.F.: Similarity metric based on siamese neural networks for voice casting. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019)

9. Gresse, A., Rouvier, M., Dufour, R., Labatut, V., Bonastre, J.F.: Acoustic pairing of original and dubbed voices in the context of video game localization. In: Annual Conference of the International Speech Communication Association (INTERSPEECH) (2017)
10. Iida, A., Campbell, N., Higuchi, F., Yasumura, M.: A corpus-based speech synthesis system with emotion. Speech communication **40**(1-2), 161–187 (2003)
11. Iosif, E., Klasinas, I., Athanasopoulou, G., Palogiannidi, E., Georgiladakis, S., Louka, K., Potamianos, A.: Speech understanding for spoken dialogue systems: From corpus harvesting to grammar rule induction. Computer Speech and Language **47**, 272–297 (2018)
12. Koch, G., Koch, G.: Siamese Neural Networks for One-Shot Image Recognition. Cs.Toronto.Edu **2** (2015)
13. Li, H., Ma, B., Lee, K.A.: Spoken language recognition: From fundamentals to practice. Proceedings of the IEEE **101**(5), 1136–1159 (2013)
14. Lu, X., Li, S., Fujimoto, M.: Automatic speech recognition. SpringerBriefs in Computer Science pp. 21–38 (2020)
15. Malik, A., Nguyen, H.: Exploring automated voice casting for content localization using deep learning. SMPTE Motion Imaging Journal **130**(3), 12–18 (2021)
16. Obin, N., Roebel, A.: Similarity search of acted voices for automatic voice casting. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**, 1642–1651 (2016)
17. Obin, N., Roebel, A., Bachman, G.: On automatic voice casting for expressive speech: Speaker recognition vs. speech classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2014)
18. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., et al.: The kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding (2011)
19. Schröder, M.: Emotional speech synthesis: A review. In: European Conference on Speech Communication and Technology (EUROSPEECH). pp. 561–564 (2001)
20. Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., et al.: The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In: Annual Conference of the International Speech Communication Association (INTERSPEECH) (2013)
21. Stylianou, Y.: Voice transformation: a survey. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3585–3588 (2009)
22. Swain, M., Routray, A., Kabisatpathy, P.: Databases, features and classifiers for speech emotion recognition: a review. International Journal of Speech Technology **21**(1), 93–120 (2018)