ISSN 1870-4069

Children Age and Gender Classification Based on Speech Using ConvNets

Humberto Pérez-Espinosa[1,2], Himer Avila-George[1,2], Juan Martínez-Miranda[1,2], Ismael Espinosa-Curiel[2], Josefina Rodriguez-Jacobo[2], Hector A. Cruz-Mendoza[2]

[1] CONACYT, Mexico

[2] CICESE-UT, Tepic, Nayarit, Mexico

{hperez, jmiranda, himerag}@cicese.mx

Abstract. In this paper, we present a study about building age and gender automatic classifiers for children at their first school years (between 6 and 11 years old). We created a speech corpus with 174 children interacting with a couple of robots in a Wizard of Oz scenario. The recorded speech was manually segmented and then characterized with low-level acoustic features. Next, we trained the classification models using a convolutional neural network architecture. Due to the complexity in the tuning process for the correct selection of the parameters used for this type of neural network, we integrated the use of a mathematical object called covering arrays to generate the set of optimal parameters for neural network architecture. Given the complexity of the classification of children speech, we obtained encouraging results. Our results indicate that it is difficult to achieve an accurate classification of children with very close ages. By grouping the subjects into two or three ages, the results improved significantly. On the other hand, the task of gender identification was less challenging, and we obtained higher classification performance measures.

Keywords: children speech, artificial neural networks, covering arrays.

1 Introduction

The automatic recognition of paralinguistic information can be useful to adapt better and personalize speech-based user interfaces. Paralinguistic phenomena can be extracted from the acoustic speech signal and then be used to identify, for instance, the speaker's identity, accent, gender, age, personality traits or emotional states. The automatic identification of this information facilitates the adaptation and personalisation of specific tasks in a human-computer interaction system or even in the assessment of the quality of the speech-based interaction between the user and the system [16].

The researchers have addressed the study of the correlation between paralinguistic information and individual's characteristics such as age and gender since the 1950s [15]. More recently, researchers have applied machine learning techniques in the construction of automatic classifiers which can recognize age and gender from adult's speech. An example of these works is [12], where the authors used Mel-frequency cepstral coefficients (MFCC) and delta regression coefficients to classify the age of Japanese speakers automatically. In [14], the authors compared different classification methods in a study of automatic classification of age group and gender. Moreover, in [13], the author refined and measured the significance of the long-term features to the age classification task.

The paralinguistic challenge organized at INTERSPEECH 2010, included two sub-challenges where the age and gender of speakers need to be identified using the "*aGender*" corpus containing 65,364 single utterances of 954 speakers with ages ranging from 7 to 80 years old [19]. For the age sub-challenge, four classes were defined: 7-14 (child), 15-24 (youth), 25-54 (adult) and (55-80) senior. The gender sub-challenge considered three categories: f (female), m (male) and x for children. This work found the discrimination of children's gender as notable difficult. [9], presented the best accuracy obtained for age classification. The authors used a fusion of different subsystems and employed Gaussian Mixture Models (GMM) and Support Vector Machines to model the tasks. In the gender sub-challenge, the best accuracy was obtained by a work that also used the fusion of different sub-systems and the classification models were constructed using SVM, Multi-Layer Perceptrons (MLP) and GMM [11]. A deeper review of the research efforts in the automatic classification of age and gender from speech, as well as the works and results obtained in the INTERSPEECH challenge, can be found in [20].

Much of the effort in the automatic recognition of age and gender using paralinguistic information has included children speech data as a single class to be differentiated from adults and elderly. Not many studies have been developed to find relevant information for age and gender classification between children at different ages. The development of speech-based interactive systems for children is equally important than those addressed for adults. Some applications of these methods include the study of child development [10], child education [23], or facilitators for autism therapy [8], to name a few.

The main contribution of this article is the classification of age and gender in children from their voices in a range from 6 to 11 years old, which is a challenging and little-explored task. Furthermore, to solve this task, we use a convolutional neural network which is a promising technique for the processing of audio signals.

2 Related Work

Only a small group of research initiatives have studied the acoustic features of children speech for the construction of automatic classifiers. One of these studies is the presented in [18], where the authors show the results of experiments in speaker recognition to identify a child in a class (30 children, similar age) and

the school (288 children, varying ages). Using the GMM and SVM approaches, the authors reported accuracy from 90%, for young children to 99%, for older children in the speaker recognition in one class, and 81% of identification rate achieved for a child in the school.

Regarding age identification in children from paralinguistic information, the work presented in [17], describes the experiments on age-group identification using the OGI Kids speech corpus [22], that contains recordings of words and sentences from 1,100 speakers. For this study, 766 speakers were chosen randomly for training and the remaining 334 for testing. The authors used GMM-UBM, GMM-SVM and i-vector systems for the construction of the classifiers. They used three age-group classes for age identification: 5-9, 9-13 and 13-16 years old. The reported results show that the GMM-UBM and i-vector systems considerably outperform the GMM-SVM system. The i-vector system applied to band-limited speech to 5.5 kHz obtained the best Age-ID performance 85.77%.

A similar study presents the work developed for the automatic recognition of gender in children using the same OGI Kids speech corpus [17]. In this experiment, the authors used GMM-UBM and GMM-SVM approaches for the identification of gender employing age-independent and age-dependent models. The authors obtained the best results when they used age-dependent models (GMM-SVM: 79.18%, GMM-UBM: 71.76%), in comparison with the age-independent models (GMM-SVM: 77.44%, GMM-UBM: 67.39%). Another relevant reported result is that the frequencies below 1.8 kHz and above 3.8 kHz are most useful for gender identification for older children, while the frequencies above 1.4 kHz are most useful for the youngest children.

In this paper, we report experiments focused on the identification of age and gender in children's speech. A main difference with the above-presented studies is that we focused on children attending to the elementary school (ages 6-11). Using a speech corpus from 174 children that we collected through an interactive scenario with a robot, our primary aim is to investigate to what extent is possible a relevant discrimination of age and gender in children aging in this age range.

3 Methodology

3.1 Speech Data Collection

Given our interest in the study of children, we decided to create a corpus of children's emotional speech. We collected children's speech during the interaction with two Lego robots. We induced different reactions in children depending on the robot's behavior. The activity that children performed consisted in giving verbal directions to the robots to go from one place to another, picking up candies and avoiding obstacles. We used two Lego Mindstorms EV3 robots, and we built a scenario in a 2x3 meters floor mat containing on its surface a route that the robots must follow, from the start to the finish position, see Figure **1**. The route has a curved trajectory comprising six depots represented as rectangles with three different colors.

There is a small bowl containing a pre-defined number of candies on each depot. The color of each depot represents the number of candies contained inside the bowl: blue depots contain bowls with two candies, orange depots have three candies in their bowls, and yellow depots hold a bowl with four candies.

Additionally, there are fourteen obstacles distributed along the route. The obstacles have the same colors as the depots to represent the number of candies lost if the robot knocks over the obstacle: blue obstacles represent the loss of two candies, orange obstacles for three candies and yellow obstacles represent four lost candies.

Fig. 1. From left to right: Child interacting with the robot, the technician operating the Text to Speech system, the technician controlling the robot's movements, the facilitator, and a friend of the child.

At the beginning of the interactive session, a facilitator explains to each child his/her mission: the primary goal is to guide the robot (using verbal commands), through the route, from the start position to the exit of the route. A second objective is to collect as many candies as possible from the bowls located in the depots. The robot needs to enter completely into the depot to collect the candies from the bowl. If the robot passes a depot without entering, the child loses the candies of that depot, given that the robot cannot go back. Additionally, the robot should avoid the obstacles located through the route. If the robot knocks over an obstacle, the candies collected so far are lost depending on the color of the obstacle.

A Wizard of Oz method has been used to generate the movements and the dialogues of the robots which are controlled by two technicians. One technician was responsible for simulating the movements of the robot using the mobile application Robot Commander App provided by Lego Mindstorms. The other technician was in charge of generating the robot's speech using a MacBook Air laptop. We used the Text To Speech engine of the OS X to synthesize the prompts spoken by the robots.

The robots used a vocabulary of 162 pre-defined sentences.

With the objective of generating speech with paralinguistic variations, the children interacted with two robots which are physically identical but have different personalities. One of the robots acts in a non-collaborative manner, ignoring some of the commands given by the child and playing some prompts to blame the child for the errors. This robot also shows a selfish behavior, keeping all the credit when it entered into stations. The other robot has a collaborative behavior and a happy mood. The utterances spoken by this robot encourage the child to get more candies. This robot is obedient and gives the child credit for his or her achievements. Children engaged in the activity and reacted emotionally to positive and negative events such as winning or losing candies, avoiding or knocking over obstacles, and being congratulated or reproached by the robot. They showed negative emotions like frustration during the session with the non-collaborative robot and showed positive attitudes and emotions like enthusiasm and joy during the interaction with the collaborative robot.

The children speech was collected using a wireless headset Logitech h600. This headset uses a USB receiver antenna with a range up to 10 m. We connected the antenna to a Dell computer with Windows 8.1 operating system that records the voice using the software Audacity v2.1.1.

A total of 174 children (78 female, 96 male), aged 6-11 (8.62 mean, 1.73 standard deviation), participated in the experiment. After finishing the recording of the interactive sessions, we manually segmented the children speech at speaker turn level. We define turn level as continuous segments in which the child is speaking uninterruptedly. It means that we take every intervention of the children (each instruction or answer to the questions made by the robot), as one speech segment. We use the software Audacity to analyze the audio recordings and cut out the parts of interest. After the segmentation, we obtained 18,674 segments. Table **1**, shows some details about the collected data.

Table 1. Minimum, maximum, average and standard deviation minutes that were used to accomplish the mission according to age of the participant.

Age	*Min*	*Max*	*Avg.Dur.*	*Std.Dev.*	*Participants*	*Female*	*Male*	*Segments*
6	05:51	22:46	13:59	03:36	32	20	12	3,739
7	08:09	20:05	13:53	03:03	21	8	13	2,714
8	07:45	21:47	12:21	02:47	23	9	14	2,473
9	08:52	17:31	11:58	02:55	30	12	18	3,111
10	05:54	19:03	11:00	02:27	41	19	22	3,697
11	06:39	16:42	10:59	02:02	27	12	15	2,940

3.2 Audio Characterization

We characterized the audio data acoustically using the software openSMILE [4]. We used the set of features proposed in [21], that is designed to reflect a broad coverage of paralinguistic information assessment.

This feature set includes the Low-Level Descriptors (LLD), listed in Table **2**. We computed these acoustic features using a frame size of 25 ms and a frame step of 10 ms. We applied a moving average filter for smoothing data contours.

We are using static acoustic features; this means that they do not model temporal phenomena, which might be discriminative for age and gender recognition[9]. For this purpose, we compute LLDs and then generate delta and double-delta regression coefficients. The result of this computation is three numbers per each of the speech sample frames. Delta and double-delta coefficients are calculated using the following equation:

$$d^t = \frac{\sum_{i=1}^{w} i * (x^{t+1} - x^{t-i})}{2\sum_{i=1}^{w} i^2,},$$

where w is the length of the frame and x^t is the signal data.

Then we calculated 39 statistical functions over the values of the LLD, its delta, and its double deltas coefficients in each frame. The 39 statistical functions that we use are: range, maximum position, minimum position, maximum mean distance, minimum mean distance, linear regression coefficient 1, linear regression coefficient 2, linear regression error A, linear regression error Q, quadratic regression, coefficient 1, quadratic regression coefficient 2, quadratic regression coefficient 3, quadratic regression error A, quadratic regression error Q, centroid, variance, standard deviation, skewness, kurtosis, quartile 1, quartile 2, quartile 3, inter-quartile range: quartile 2-quartile 1, inter-quartile range: quartile 3-quartile 2, inter-quartile range: quartile 3-quartile 1, percentile 95.0, percentile 98.0, zero crossing rate, number of peaks, mean peak distribution, peak mean, peak mean mean distribution, arithmetic mean of the contour, absolute mean, quadratic mean, nz absolute mean, nz quadratic mean, nz geometric mean, and nnz.

The result of this procedure is a set of 6,552 attributes for each single audio sample. After an experimentation stage comparing several feature selection methods, the method with the best results was *Relief Attribute*. We used this evaluation method as implemented in Weka [6]. The method showed the best accuracy rates when we took the 350 best-ranked attributes. We selected these features from the original feature set of 6,552 attributes to obtain the best descriptors and reduce the dimensionality of the attributes vector.

3.3 Convolutional Artificial Neural Networks

The convolutional artificial neural Networks (ConvNets), are biologically inspired by the Human Visual System "HVS" and its hierarchal architecture [7]. In these networks, the weights are shared across time or space. Neurons with the same weights are applied on input patches of the previous layer at different segments of the input data. In this way, it is achieved a degree of *translational invariance*, by computing feature maps [5], that allows the networks to learn patterns and reuse them in different space or time context.

Table 2. Set of acoustic features.

LLD	***Number of features***
F0 via autocorrelation function	117
F0 envelope	117
MFCC [0-12]	1,521
Spectral band energy (0-250 Hertz)	117
Spectral band energy (0-650 Hertz)	117
Spectral band energy (250-650 Hertz)	117
Spectral band energy (1000-4000 Hertz)	117
Mel-spectrum [0-25]	3,042
Spectral centroid	117
Spectral Flux	117
Spectral maximum position	117
Spectral minimum position	117
Spectral roll-off point (25%)	117
Spectral roll-off point (50%)	117
Spectral roll-off point (75%)	117
Spectral roll-off point (90%)	117
Logaritmic energy	117
Zero-crossing rate	117
Voicing probability	117
Total	6552

Therefore, ConvNets are useful when the inputs samples are statistical invariants, that is, the input samples contain the same kind of information, and it does not change on average across time or space. In a ConvNet, instead of having stacks of matrix multiply layers, there are stacks of convolutions. Currently, pattern recognition systems based on convolutional networks are among the best performing systems. This type of network architecture typically has five, six or seven layers [3]. ConvNets are structured by stacking up convolution layers. Usually, at the top of the structure there are fully connected layers, and at the end of the structure, there is a classifier.

Some critical parameters need to be configured in a ConvNet. For instance, *stride*, is the number of shifted features each time the filter moves. The strides between layers are used to reduce the dimensionality and to increase the depth of the neural network. Each layer in the structure is called a *feature map*. There could be more than one feature map, for instance, if an image has the channels R, G, and B, these three channels (K-size = 3), can be handled as individual features maps. In our particular case, we are handling only one channel because the audio recording is mono-channel.

When the shifting filter does not go beyond the edge of the feature vector, it is called valid padding. If the filter goes off the edge of the feature vector and therefore the output map size is the same size than the input map, it is called *same padding*. The pooling layers take all the convolutions in a neighborhood and combine them instead of skipping one in every two convolutions. Max pooling takes a small neighborhood around every point in the feature map and computes the maximum of all the responses around it. Given that the convolutions are done on lower stride, the structure becomes more expensive to calculate, and there are more parameters to tune. The average pooling, instead of taking the max, takes an average over the window of features around a particular location.

We implemented a typical architecture for a ConvNet.

It has two alternated layers of convolution and pooling, followed by a fully connected layer and a classification layer at the end. Fig. **2**, shows the network structure.

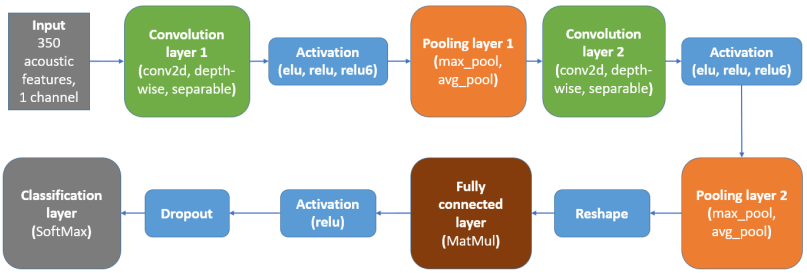

Fig. 2. Convolutional neural network architecture used for the experiments presented in this paper. The different options that we tested at each layer are shown in parentheses.

For the implementation of the convolutional neural network, we used the software TensorFlow [1]. An important characteristic of this tool is its flexible architecture which allows deploying computation to one or more CPUs or GPUs in a desktop, server, or mobile device using the same API. Google released TensorFlow under the Apache 2.0 open source license.

3.4 ConvNet Parametrization

Below we present the list of variables that needs to be tuned for the accurate training of the ConvNet and the corresponding values that we selected for testing. The $v\#$, stands for the identifier of the variable.

- **v1 - Num_hidden:** a positive integer that represents the depth of the convolution layers. It can have nine values, from 32 to 96 with steps of eight between values.
- **v2 - Num_epochs:** a positive integer that limits the number of steps that the validation set is evaluated. It can have seven values, from 10 to 70 with steps of 10 between values.
- **v3 - Learning_rate:** a float that represents the magnitude of the update per each training step. Decay once per epoch. It can have seven values, from 0.0007 to 0.003 with steps of 0.000383 between values.
- **v4 - Batch_size:** a positive integer that indicates the percentage of the training data that are feed at each iteration. It can have seven values, from 5 to 11 with steps of one between values.
- **v5 - Eval_batch_size:** a positive integer that indicates the percentage of the validation data that is feed at each iteration. It can have seven values, from 8 to 32 with steps of four between values.

v6 - **Patch_size** a positive integer that indicates the size of the window that slides across the feature vector. It could also be bidimensional, in our case it is unidimensional. It can have five values, from 3 to 7 with steps of one between values.

v7 - **Depth** a positive integer that represents the depth of the second convolution layer. It can have nine values, from 16 to 32 with steps of four between values.

v8 - **Strides** a list of integers that represents the number of features that are shifted by the sliding window for each dimension of the input vector each time the filter moves. It can have three values, from 1 to 3 with steps of one between values.

v9 - **K-size** an integer that represents the size of the dimension of the input. It can have three values, from 1 to 3 with steps of one between values.

v10 - **Partition** an integer that represents the distribution of the data for the train, validation, and test sets. It can have three values, from 1 to 3 with steps of one between values. The tested partition were 1 = 33/33/33, 2 = 40/40/20 and 3 = 70/20/10.

v11 - **Padding** If the filter goes off the edge of the feature map or not. The two possible values for this parameter are same and valid. It can have two values, from 1 to 2 with steps of one between values.

v12 - **Optimizer:** functions to compute gradients for a loss measure and apply gradients to variables. It can have two values, from 1 to 2 with steps of one between values. The possible values of this parameter are 1 = *GradientDescentOptimizer*, 2= *AdamOptimizer*.

v13 - **Activation:** functions that provide nonlinearities. It can have two values, from 1 to 2 with steps of one between values. 1 = continuous but not everywhere differentiable functions (*relu6*) 2= smooth nonlinearities functions (*elu*).

The methodology for setting the values of the parameters of an ANN is based on the study of the effect on the quality of the solution, which is caused by the interaction between the variables of the experiment.

The tuning process of the parameters of the ANN was done using a mixed-level covering array denoted by $MCA(N; t, k, (v_1, \ldots, v_k))$. An MCA is a $N \times k$ matrix in which the entries of the ith column arise from an alphabet of size v_i; additionally, each column $i(1 \leq i \leq k)$ contains only elements from a set S_i with $|S_i| = v_i$, and the rows of each $N \times t$ subarray cover all t-tuples of values from the t columns at least once.

We use a $MCA(64; 2, 9^1 7^4 5^2 3^3 2^3)$ to tune the parameters of the ANN, this MCA represents the experimental plan. The MCA was constructed using the simulated annealing algorithm reported in [2]. The experimental plan is composed of sixty-four experiments for each of the ages groupings tested.

4 Results

For the experiment implementation, we used the TensorFlow framework installed on a Linux server equipped with 72 Intel Xeon processors, 64GB RAM, and a Tesla K20 GPU accelerator. The Linux distribution for this server is CentOS.

The Table **3**, shows the results of the classification experiments including precision, recall, and F1-score. These metrics reach its best value at 1; this is, the closest to 1, the most accurate is the classification. To measure the classification performance we used three data subsets: training, validation, and testing. The results showed in the tables are the ones obtained with the test set. The speech segments of each child belong only to one of the three data subsets to ensure the training of speaker-independent models.

As we can see in table **3**, we obtained the best results when we classified two classes of age and two classes of gender. For age and gender classification we got comparable results to the ones reported in the literature. However, our results do not surpass the obtained by [17], using the OGI Kids speech corpus. It is important to say that we are working with a narrower age group where it is expected to have less variability in the acoustic properties of the children speech. Another important difference is that our data were recorded in a noisier environment. We also built classification models for three groups of ages and all the six included ages without grouping. As reported before [19], we found these task tough, obtaining a poor classification performance.

The Table **4**, shows the best combinations of parameters for each of the classification tasks. We can see that from the 64 tested combinations, experiment 60 had the best performance for two tasks (2 age classes, and two gender classes). On the other side, the combinations 30 and 7 had the worst performance of the two task each one. From these results, we can have a better idea about what are the most recommendable parameters for the ConvNet for these tasks. We can see for example that to obtain a good classification the parameter v8 (strides), should not be set to 1, or that the ConvNet has a better performance when parameter v13 (activation), is set to 2.

5 Conclusions

Children's age classification is a challenging task; we were able to obtain a good classification performance grouping the data in two classes. However, we got poor results when we tried to classify the children per years old. We consider that the results obtained classifying by gender were good in comparison with results previously reported by other authors. We found that covering arrays are a useful tool to parameterize a ConvNet, given that we were able to select an adequate combination of parameters for our classification tasks. The obtained results encourage us to keep researching on the combination of deep neural networks structures and covering arrays to parameterize and solve classification tasks related to paralinguistic phenomena.

Table 3. Gender and age classification performance for different ranges. 2 - Classes (6,7,8 vs 9,10,11), 3 - Classes (6,7 vs 8,9 vs 10,11), 6 - Classes (6 vs 7 vs 8 vs 9 vs 10 vs 11). The table shows the maximum, minimum, mean and standard deviation of the evaluation metrics for the 64 experiments' results.

Number of classes	*Max*	*Min*	*Mean*	*StdDev*
	F1-score			
2 Classes Age	**0.71**	0.36	0.56	0.11
3 Classes Age	**0.50**	0.18	0.38	0.11
6 Classes Age	**0.34**	0.04	0.16	0.09
2 Classes Gender	**0.71**	0.37	0.52	0.10
	Precision			
2 Classes Age	**0.72**	0.27	0.58	0.10
3 Classes Age	**0.50**	0.12	0.38	0.11
6 Classes Age	**0.33**	0.04	0.17	0.10
2 Classes Gender	**0.71**	0.28	0.54	0.10
	Recall			
2 Classes Age	**0.72**	0.49	0.59	0.07
3 Classes Age	**0.53**	0.31	0.43	0.07
6 Classes Age	**0.35**	0.12	0.22	0.09
2 Classes Gender	**0.71**	0.28	0.54	0.10

Table 4. Best and worst experiments parametrization for ConvNet

Classes	*Exp*	*v1*	*v2*	*v3*	*v4*	*v5*	*v6*	*v7*	*v8*	*v9*	*v10*	*v11*	*v12*	*v13*
					Best result									
2 Classes age	60	48	30	0.0022	11	24	6	32	3	2	3	1	2	2
3 Classes age	41	72	30	0.0019	9	8	5	28	2	2	2	2	2	2
6 Classes age	37	48	70	0.0026	7	28	5	24	3	3	2	1	2	2
2 Classes gender	60	48	30	0.0022	11	24	6	32	3	2	3	1	2	2
					Worst result									
2 Classes age	30	40	50	0.0026	6	28	7	20	1	1	1	1	2	1
3 Classes age	30	40	50	0.0026	6	28	7	20	1	1	1	1	2	1
6 Classes age	7	64	60	0.0030	8	12	4	28	1	2	2	1	2	1
2 Classes gender	7	64	60	0.0030	8	12	4	28	1	2	2	1	2	1

Acknowledgements. This research work has been carried out in the context of the "Cátedras CONACyT" program funded by the Mexican National Research Council (CONACyT). This work was financed by CONACYT under the Thematic Networks of Research program (Thematic Network on Language Technologies Ref. 260178, 271622).

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., et al: TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org 1 (2015)
2. Avila-George, H., Torres-Jimenez, J., Gonzalez-Hernandez, L., Hernández, V.: Metaheuristic approach for constructing functional test-suites. IET Software 7(2), 104–117 (2013)
3. Bengio, Y.: Learning deep architectures for AI. Foundations and trends® in Machine Learning 2((1)), 1–127 (2009)

4. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the international conference on Multimedia. pp. 1459–1462. ACM (2010)
5. Gens, R., Domingos, P.: Deep symmetry networks. In: Advances in neural information processing systems. pp. 2537–2545 (2014)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. SIGKDD explorations newsletter 11((1)), 10–18 (2009)
7. Hassan, M., Khalifa, O., Talib, A., Abdulla, A.: Unconstrained facial recognition systems: a review. Asian Journal of Applied Sciences 3((02)) (2015)
8. Ketterl, M., Knipping, L., Ludwig, N., Mertens, R., Rahman, M., Ferdous, S., Ishtiaque Ahmed, S., Anwar, A.: Speech development of autistic children by interactive computer games. Interactive Technology and Smart Education 8(4), 208–223 (2011)
9. Kockmann, M., Burget, L., Černocký, J.: Brno university of technology system for interspeech 2010 paralinguistic challenge. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010). vol. 2010, pp. 2822–2825. International Speech Communication Association (2010)
10. Laplante, J., Michaud, F., Larouche, H., Duquette, A., Caron, S., Letourneau, D., Masson, P.: Autonomous spherical mobile robotic to study child development. IEEE Trans. on Systems, Man, and Cybernetics (2005)
11. Meinedo, H., Trancoso, I.: Age and gender classification using fusion of acoustic and prosodic features. In: INTERSPEECH. pp. 2818–2821. Citeseer (2010)
12. Minematsu, N., Sekiguchi, M., Hirose, K.: Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers. In: Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on. vol. 1, pp. I–137. IEEE (2002)
13. Muller, C.: Automatic recognition of speakers' age and gender on the basis of empirical studies. In: Proceedings of the Interspeech,Pittsburgh, Pennsylvania. pp. 1–4 (2006)
14. Muller, C., Wittig, F., Baus, J.: Exploiting speech for recognizing elderly users to respond to their special needs. In: Proceedings of the Eurospeech, Geneva, Switzerland. pp. 1305–1308 (2003)
15. Mysak, E.: Pitch and duration characteristics of older males. Journal of Speech & Hearing Research (1959)
16. Pérez-Espinosa, H., Martínez-Miranda, J., Espinosa-Curiel, I., Rodríguez-Jacobo, J., Avila-George, H.: Using acoustic paralinguistic information to assess the interaction quality in speech-based systems for elderly users. International Journal of Human-Computer Studies 98, 1–13 (2017)
17. Safavi, S., Russell, M., Jancovic, P.: Identification of age-group from children's speech by computers and humans. In: INTERSPEECH. pp. 243–247 (2014)
18. Safavi, S., Najafian, M., Hanani, A., Russell, M.J., Jancovic, P., Carey, M.J.: Speaker recognition for children's speech. In: INTERSPEECH. pp. 1836–1839 (2012)
19. Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Muller, C.A., Narayanan, S., et al: The interspeech 2010 paralinguistic challenge. In: InterSpeech. vol. 2010, pp. 2795–2798 (2010)
20. Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., Narayanan, S.: Paralinguistics in speech and language—state-of-the-art and the challenge. Computer Speech & Language 27(1), 4–39 (2013)

21. Schuller, B., Steidl, S., Batliner, A., et al: The interspeech 2009 emotion challenge. In: Interspeech. pp. 312–315 (2009)
22. Shobaki, K., Hosom, J., Cole, R.: The OGI kids' speech corpus and recognizers. pp. 564–567. ICSLP (2000)
23. Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., Vuuren, S.V., Weston, T., Zheng, J., Becker, L.: My science tutor: A conversational multimedia virtual tutor for elementary school science. ACM Transactions on Speech and Language Processing (TSLP) 7(4), 18 (2011)