New proposed Solution for speech recognition without labeled data: Tutoring System for Children with Autism Spectrum Disorder

Amel ZIANI*,[1][0000-0003-3433-0596], Amine ADOUANE[2][0000-0002-7582-4202],

Mohamed Nassim AMIRI[3], Sabiha SMAIL[3]

[1] Chadli Benjedid University, Computer science department, El tarf, Algeria.
[2] Benyoucef Benkhedda University, Computer science department, Algiers, Algeria.
[3] Algiers, Algeria.
*/a.ziani@univ-eltarf.dz,amine.adouane@univ-alger.dz

Abstract. Children diagnosed with Autism Spectrum Disorder (ASD) face challenges in understanding situations, verbal communication, and social interactions. Autism can manifest differently in each child, and it can be characterized by various degrees of severity. Some common behaviors observed in children with ASD include poor skills, repetitive behaviors, delayed speech, reasoning difficulties, narrow interests, and challenges with social interactions and communication, such as recognizing social cues. As every child with ASD has unique educational needs, there is no universal solution for treating the condition. This paper aims to introduce an adaptive educational system that will help children with ASD acquire new skills and improve their communication abilities, enabling them to better integrate into society. The proposed system will be based on therapist-researched and -analyzed activities that utilize speech recognition technology. To address the resource requirements of labeled datasets, we propose a new approach that leverages generative adversarial networks (MelGAN) to produce responses that closely resemble a child's voice. This allows for the comparison of the generated response with the correct answer using similarity metrics. The system was tested on Algerian children with ASD who speak Algerian dialect, and the results were promising and this can open a new direction for developing educational systems that do not rely on labeled datasets.

Keywords: Autism spectrum disorder (ASD), Arabic speech recognition, Generative adversarial networks, MelGAN-VC Generators, wav2vec.

1 Introduction

Autism is considered as a neurological disorder that is caused by slow brain development which results in difficulties of communication with others, to learn, to express their emotions, to adapt to a new environment, etc. Autistic people are not capable to interact properly with their surroundings so they keep a distance from them [1]. Hence, they can not develop their interaction and communication skills which results on not being able to analyze the behavior and intent of others or to adapt to others routine [2]. Both Autism Spectrum Disorder (ASD) [3] and Attention Deficit Hyperactivity Disorder (ADHD) [4] are categorized as types of autism and they have common symptoms and special way of communication. This is why the life of autistic children is a tiresome for them and their families where they face challenges from early childhood through the rest of their lives. Nevertheless, they require special tutoring, special

schools, and an adapted way of treatment and communication. As all the therapists agreed that there is no cure for autism currently, but some therapies can improve their conditions and ameliorate their behavior and communication such as speech therapy, emotion-focused therapy and behavioral therapy. The rate of autism is augmenting per day for unknown reasons, the authors in [5] have made a review that shows the number of children with autism per 10,000 children in developed countries. And in Arab countries; Qatar: 151.2 per 10,000, United Arab Emirates: 112.4 per 10,000, Oman: 107.2 per 10,000, Bahrain: 103.3 per 10,000 and Saudi Arabia: 100.7 per 10,000[1].

A lot of applications that are based on advanced techniques of artificial intelligence [6], machine learning "ML" [7] and internet of things "IoT" [8] have proved their capabilities in enhancing their daily lives as mentioned in the work of [5]. The real time monitoring and tutoring can ensure children's timely therapy what can integrate them into the core part of the society. Thus, the processing power of AI and ML is needed to achieve that.

Some review works have been carried out, which cover the detection and intervention of autistic individuals [9, 10, 11, 12, 13, 14, 15, 16]. The author of [17], reviewed eleven virtual game-based research works that could help autistic children. He covered games related to education, entertainment, health and simulation which can run on different devices such as computers, smartphones, and touch-sensitive displays.

The IA Based monitoring or teaching systems use basically the speech recognition techniques. These latter need to be trained on labeled speech data [18, 19, 20, 21]. Which is considered as a drawback of these models, that require a lot of labeled data to perform well that is usually only available for English and a few other languages. Therefore, purely supervised training is impractical for the vast majority of the 7,000 languages spoken around the world [22] which is why there has been a lot of interest in how to better use unlabeled speech data [23, 24, 25].

The problem of unavailability of labeled data in Arabic language becomes the biggest hindrance for Arabic researchers in all fields. Therefore, we had to find a solution to avoid building datasets, as it is time and resource consuming and becoming increasingly expensive for using advanced machine learning techniques, such as deep learning, which require huge datasets.

Therefore, our proposed solution is based on generative adversarial networks that are used in deepfake and other applications to generate images, audios, and texts. The generative adversarial network (GAN) was first introduced by [26]. The essential part of GAN is to establish a game between two networks, namely the generator and the discriminator. After sufficient training, the generator obtains the ability to generate the samples whose distribution resembles the training data. MelGAN-VC was proposed in [27], as a voice conversion method that relies on non-parallel voice data and is able to convert audio signals of arbitrary length from a source voice to a target voice. Voice conversion (VC) is a technique used to change the perceived identity of a source speaker to that of a target speaker, while keeping the linguistic information unchanged. It has many potential applications, such as generating new voices for Text-To-Speech (TTS) systems [28], dubbing in movies and video games, speech assistance [29, 30] and speech enhancement [31, 32, 33]. This technique involves creating a matching function between the speech features of two or more speakers.

The tutoring system consists of asking and evaluating the autistic child to help him gain some information and skills. The proposed approach aims at generating a new answer of the autistic child based on the correct answer from the tutor to evaluate the child while conserving all the peculiarities of the source voice by using MelGAN-VC. Just to be able to compare the generated correct answer and the child's input answer by equating the features of the two vectors. This method will allow us to compare the characteristics of only two sounds from the same person (the real answer and the generated one), which will be easier than classifying the response for speech recognition which requires large sets of labeled data for Arabic autistic children. Thus, we were able to limit the comparison to two voices and avoid using classical speech recognition techniques.

2 Proposed architecture of the tutoring system for the autistic children

The interaction modules were designed to capture the children's voice after the question on a smart tablet. A qualitative study has been conducted through multiple sessions with some therapists from the autism center to determine the module that could better develop and elevate the children's learning. From the study, three important interactions were judged essential to this system as shown in Fig. 1**.** Diagram of interactions between the system and the autistic child..

[1] https://worldpopulationreview.com/country-rankings/autism-rates-by-country

This system is developed to help the tutors and the parents in the teaching of their autistic children, where this learning starts with the basic levels until they learn everything. The teaching protocol is inspired from the applied behavior analysis ABA protocol used in multiple autism centers by the speech therapists [34].

- The first interaction (from the system to the child): The system provides a series of activities for different levels in order to evaluate the autistic child using the voice of his therapist to make him comfortable.
- The second interaction (from the child to the system): the autistic child reacts to the question by a vocal answer. In between each question, some interval time were given to allow the children to answer or at least provide some response to the question.
- The third interaction is the evaluation (from the system to the child): The answer of the child is being processed by the system to evaluate it by comparing it to the correct answer and then displays the result to the autistic child.

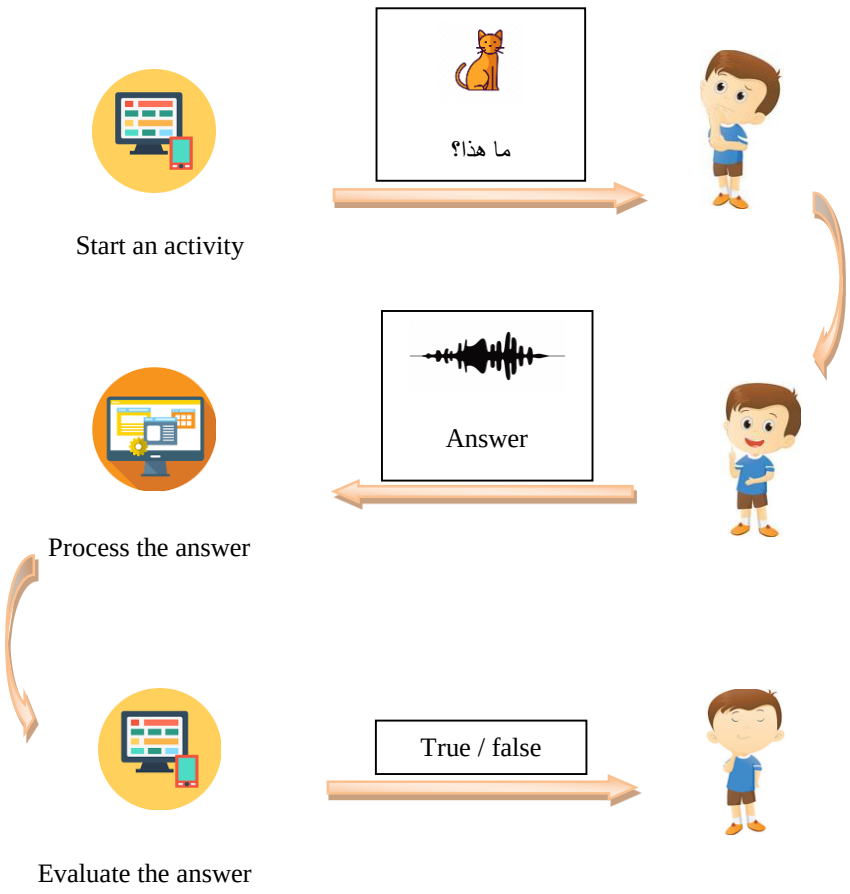


Fig. 1. Diagram of interactions between the system and the autistic child.

The vocal's answer may contain many peculiarities, such as stuttering, repetition, mispronunciation, etc. Therefore, the nature of this audio depends on the child's communication capabilities and his level of autism, and this is the main reason of not being able to compare it with the audio of the correct answer of a normal person. Even the speech recognition models for Arabic language can face difficulties to recognize the words in such audios due to the lack of Arabic autistic vocals datasets.

The proposed approach to resolve this dilemma, is to use the generative adversarial networks to change the perceived identity of a source speaker into that of a target speaker, while keeping the linguistic information unchanged. Fig. 2 explains the overall architecture.

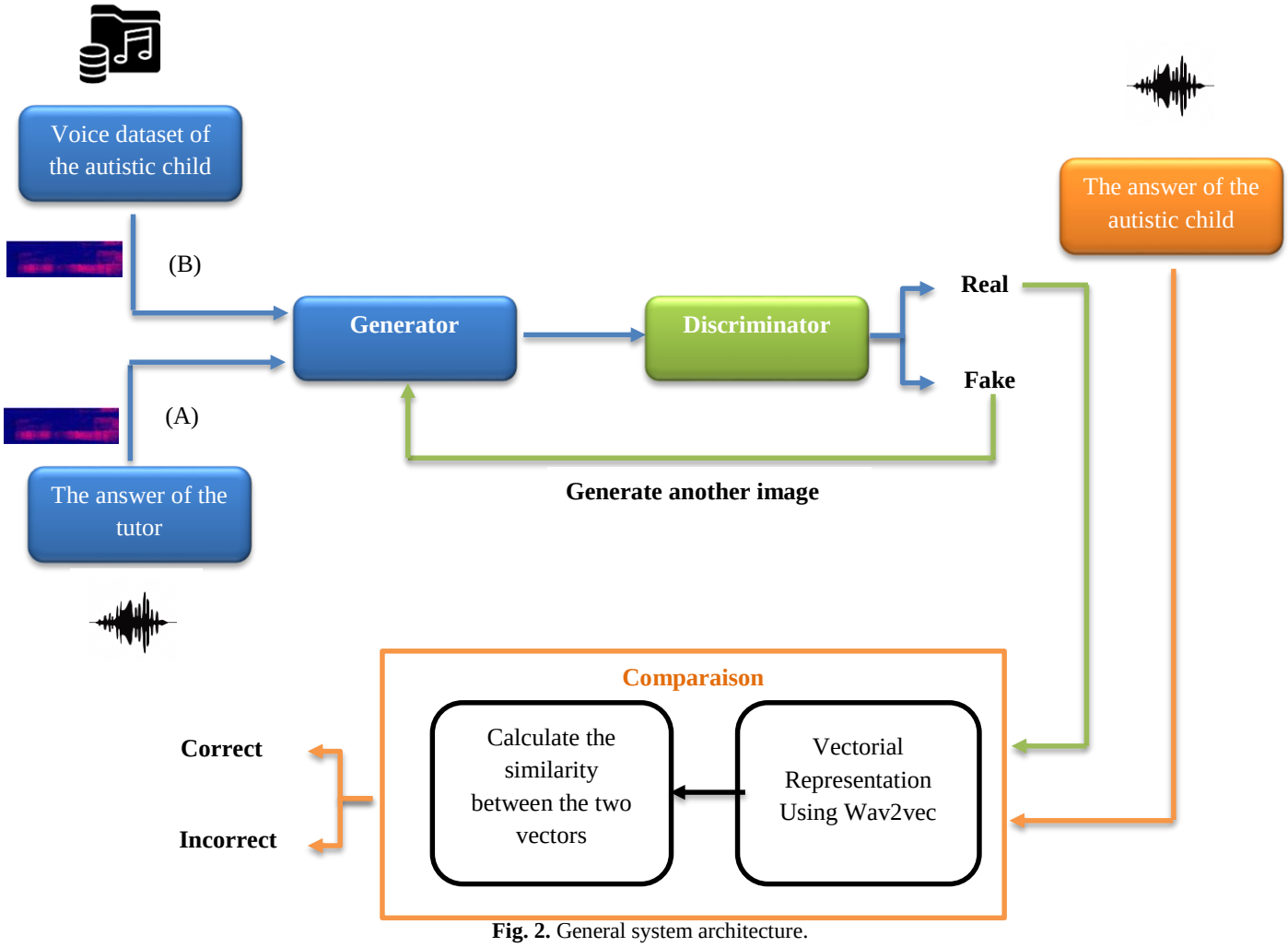


Fig. 2. General system architecture.

2.1 Dataset

Our approach has been proposed to alleviate the lack of Arabic datasets problem as mentioned above. Hence, we had to construct an unlabeled dataset that contains voices of the autistic children and that also can be collected during the use of the proposed system. This dataset will be used to feed the generator to learn the autistic child voice and its peculiarities. This dataset is composed of one thousand audio files with the extension ".wav" and with five different voices, two hundred audio files for each child.

Another small dataset that contains questions and answers from the tutor was used to test the system (Table 1):

Table 1. Example of questions and answers.

English translation	Answers	English translation	Questions
Red	أحمر	What is this color?	ما هذا اللون؟
Sea	بحر	What do you see in this picture?	ماذا ترى في هذه الصورة ؟
Lion	أسد	What is this animal?	ماهو هذا الحيوان؟

2.2 Audio pre-processing

Before the training phase of the generator, a conversion phase is needed to convert the audio ".wav" into a spectrogram ".jpeg", then this latter will be normalized between [-1, 1] so that they look like the activation function TanH() of the generator.

2.3 The generator

The proposed system is based on generative adversarial networks or GANs, which are considered as unsupervised learning algorithms. Among several variants of GAN we have opted for the MelGAN-VC architecture [27] because it is one of the few GANs that are dedicated for speech to speech. To support our choice for this variant, we will describe the chosen generator and discriminator architectures for mel-spectrogram inversion. We will explain the core components of the model and detail all the steps from splitting the spectrogram to the final generation of the audio. Fig. 3 shows the generator architecture.

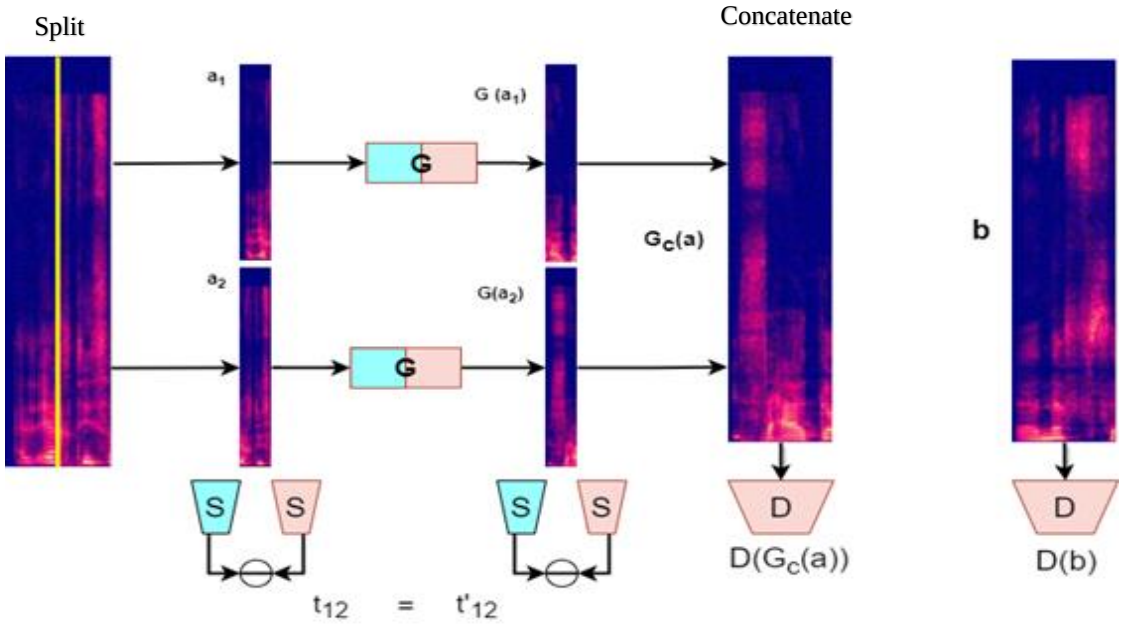


Fig. 3. MelGan-VC architecture.

This method allows speech conversion using non-parallel speech data and is capable of converting audio signals of arbitrary length from a source voice to a target voice. In our case, we use it to convert the voice of the correct answer to the voice of the autistic child.

The architecture of this GAN is composed of three neural networks, as follows:

- The generator *G*, its goal is to produce artificial data similar to real data.
- The discriminator *D*, aims to distinguish the data generated by the generator *G* from the training data.
- The Siamese network *S*, allows to keep the speech information during the translation process without sacrificing the possibility to flexibly model the target speaker's style.

After converting the audio to a spectrogram, the latter will be divided to two parts and will be fed to the generator *G*. The generated samples will be concatenated again and transmitted to the discriminator *D*. At the beginning of the learning process the discriminator classifies the generated image *D(G(input))* as false and the real image *D(correct answer)* as true. Following the iterations the generator *G* succeeds in making deceive the discriminator *D* by generating spectrograms similar to the real ones, therefore the discriminator classifies the generated sample *D (G (input))* in true and real *D (correct answer)* in false. And the Siamese network *S* is used to preserve the linguistic information transformed by *G*.

Architecture of the generator. The generator is based on U-Net architecture, a convolutional neural network (CNN) method, first proposed by Olaf Ronneberger, Phillip Fischer and Thomas Brox in 2015, with the suggestion of better segmentation of biomedical images [35].

The architecture resembles to a 'U' which justifies its name, it consists of descending and ascending sections. In this architecture, we opted for three descending and three ascending sections, as shown in the figure below (see Fig. 4).

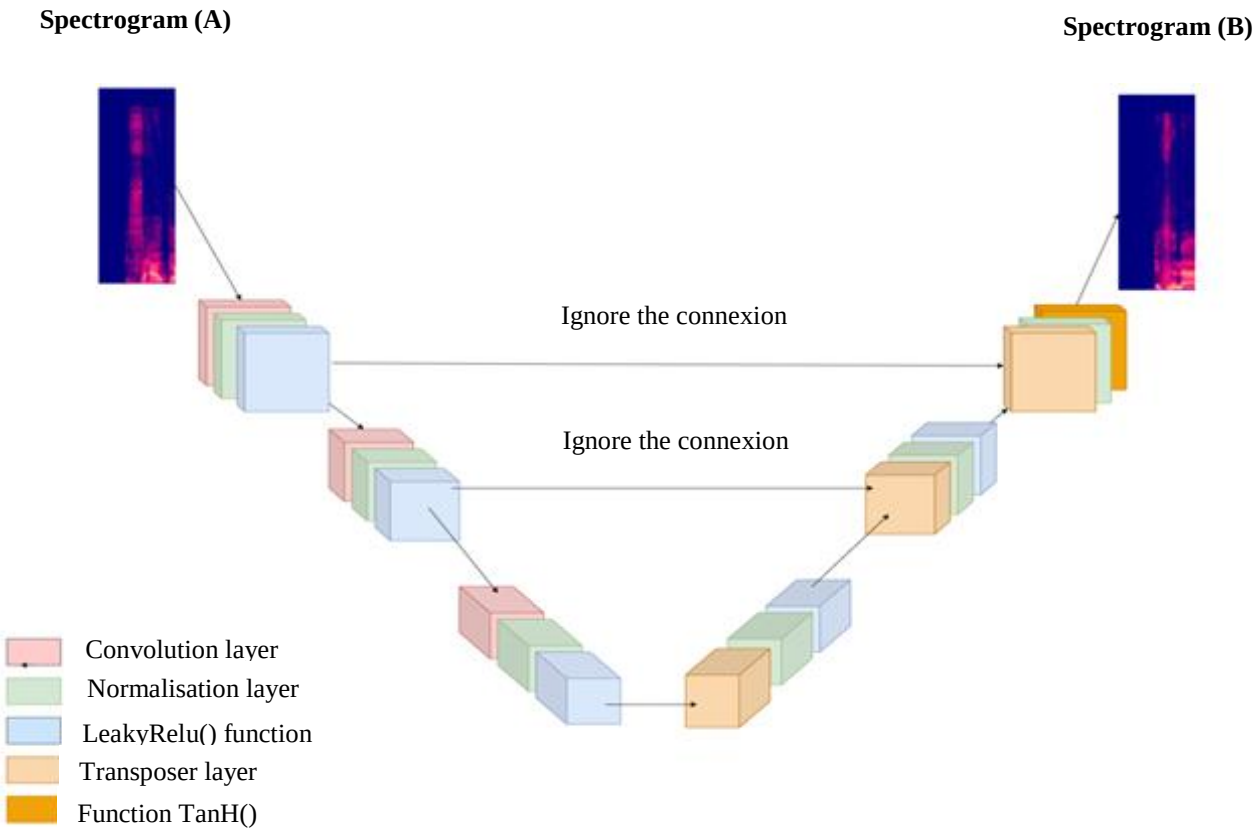


Fig. 4. The architecture of the generator.

- The first three descending sections each contain three layers, convolution layer, normalization layer and *LeakyRelu ()* as activation function.
- In the second part of the generator architecture, it is the three ascending sections. The first two sections also consist of three layers, normalization layer, activation function *LeakyReLu()*, and transpose-convolution layer.
- And for the last section we change the activation function to *TanH()*.

Convolution layer. This layer is most often used in convolutional neural networks (CNN). Convolution is an ordered process in which two sources of information are interleaved, it is an operation that transforms features into other features. Convolution has long been commonly used in image processing to blur and sharpen images, but it is also used to perform other operations. (For example, to enhance edges and embossing).

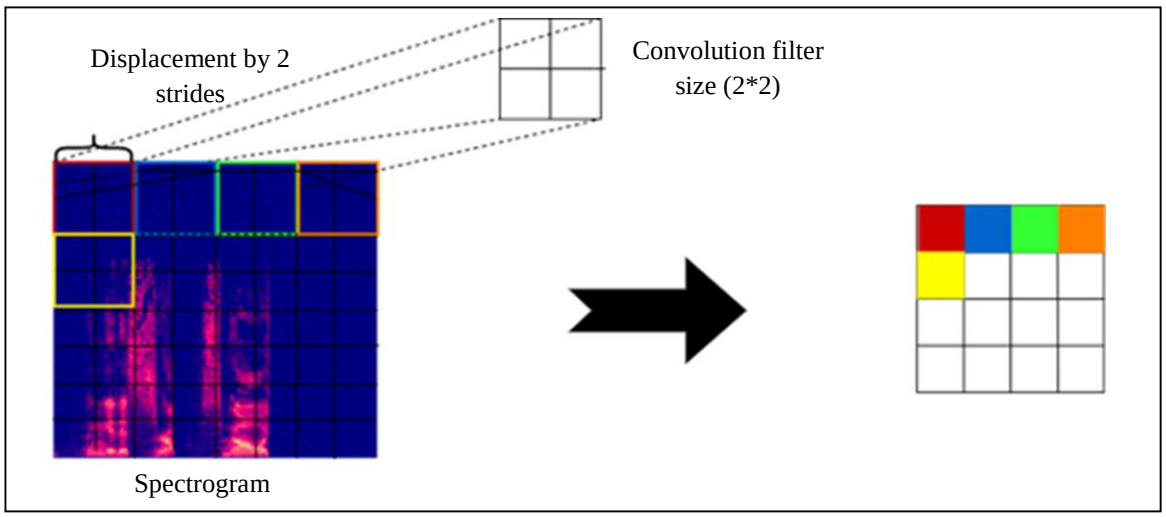


Fig. 5. The architecture of a convolutional layer.

Before applying this layer, the image (spectrogram) is in the form of a matrix of pixels of size n×m, it has three dimensions of depth to represent the fundamental colors (Red, Green, Blue). First, we define the size of the filter 2*2. This filter moves progressively from the left to the right with two steps (Stride =2) until it reaches the end of the image. This convolution calculation allows us to obtain an image as an output of smaller size than the initial one, so a (n÷2)×(m÷2), as shown in Fig. 5.

Normalization Layer. Batch normalization is a technique for improving the speed, performance, and stability of artificial neural networks. Batch normalization was introduced in [36] and is used to normalize the input layer by adjusting and scaling activations, it speeds up training, in some cases halving the epochs or better and provides some regularization to reduce generalization error.

Training networks with dozens of layers in deep learning can be very sensitive to the weight of the network. A possible reason for this difficulty is that the distribution of inputs to the deep layers of the network can change after each batch size when the weights are updated. This may cause the learning algorithm to chase a moving target forever. This change in the distribution of inputs to the layers of the network is referred to as an 'internal covariate shift' [37].

The LeakyReLu () function. An activation function is applied to each output of the convolution layers, this operation is often called a correction layer. In this model, the *LeakyRelu* function was chosen, which was proposed to allow the gradient to propagate, even when the data are negative. This allows to reactivate potentially lost (and useful) weights that the ReLu would have left at 0 [37], as it is shown in Fig. 6.

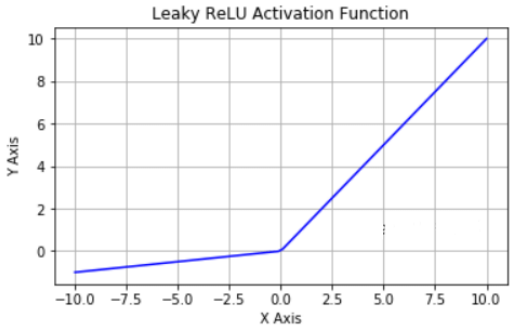


Fig. 6. Leaky ReLu activation function ([37]).

The equation of this function is : $$y = f(x) = \begin{cases} ax, si\ \ x < 0 \\ x,\ \ si\ \ x > 0 \end{cases} \quad (1)$$

Telle que a = 0.01.

Transposed convolution layer. Transposed convolutions are used in the three bottom-up sections to generate an output feature map that has a larger size than the input feature map. Thus, restoring the original (pre-convolution) image size.

The parameters that we set for this layer can be found in the table below (Table 2).

Table 2. The parameters of the transposed convolution.

Parameters		
Padding	**Stride**	**Filter**
0	2	2×2

TanH() activation function. This activation function is used only in the last ascending section, the hyperbolic tangent function is similar to the sigmoid function :

- Sigmoid gives a result between 0 and 1.
- Tanh gives a result between -1 and 1.

The equation of this function shown in Fig. 7 is:

$$y = \tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (2)$$

This feature allows the model to learn to saturate and cover the color space of the training distribution more quickly.

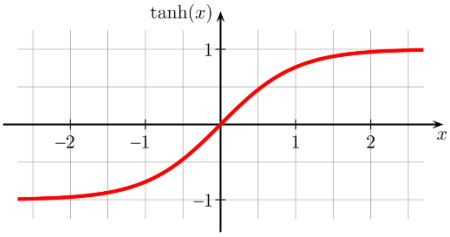


Fig. 7. Tanh() activation function [37].

The discriminator. During training, the discriminator is confronted with both real images and images from the generator. It is in charge of identifying the source of each image [38, 39, 40, 41, 42]. For this purpose, we have used the architecture shown in the figure below (Fig 8). This architecture has as input an image (spectrogram), it is either generated by the generator or it is a training image. This spectrogram goes through three sections and each section is composed of three layers: convolution layer, normalization layer and the activation function LeakyReLu() seen previously. At the end of these three sections we will have as a result a matrix with size x×y×z that will be resized as a linear vector of size (1, x×y×z), then this last one will be connected with another linear layer of size 1×1 that represents the prediction with the intention of knowing if the input result is true or false.

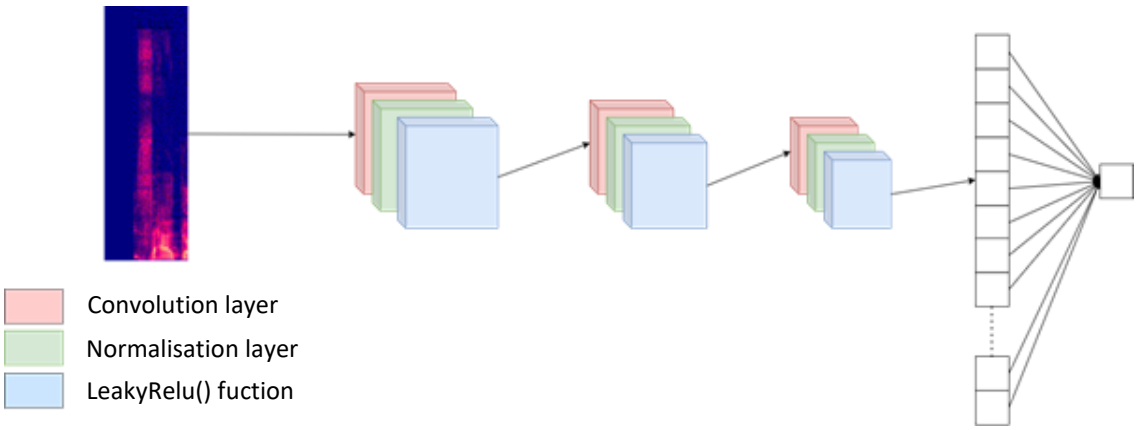


Fig. 8. Architecture of the discriminator.

Architecture of the Siamese network. Bromley D et al. introduced the Siamese network in the 90's, to answer signature verification questions. The principle is to use two networks using shared weights (same weights) to learn on the signature image, in order to verify if one signature matches another (if the result of the first network is close to the result of the second). This architecture is very interesting because this network allows learning with very little data [43].

The goal of a Siamese network is to learn the similarity between two images. Its architecture is presented in the figure below (Fig. 9).

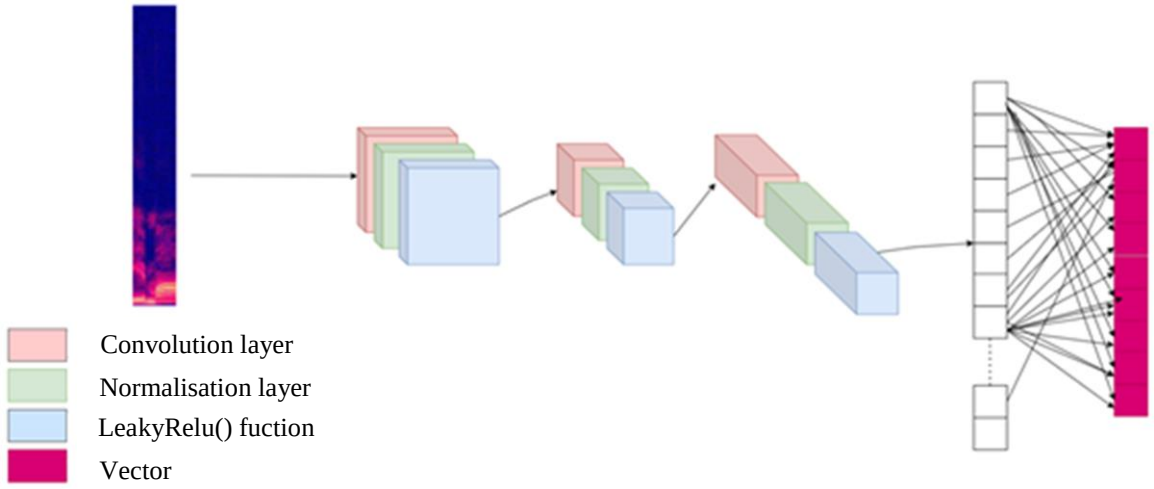


Fig. 9. Siamese network architecture.

This architecture is similar to that of the discriminator, but the only difference is that they do not have the same type of output. The Siamese network has as output a linear array of size (1,128). We will use it to compute the similarity before and after the *G* transformation, at the input of the generator a_1 (speaker) and at the output of the generator *G* (a_1), in order to preserve the linguistic data.

The training. The three models of the generator, discriminator and Siamese network are trained together. Firstly, we start by splitting the spectrogram *A* into two parts a_1 and a_2 in order to pass them to the input of the generator *G* one after the other, from which we obtain two generated results *G* (a_1) and *G* (a_2). These two latter will be concatenated to get the real generated result *G(A)*. Then *G(A)* and another training spectrogram *B* will be passed to the input of the discriminator, the results of the latter *D (G(A))* and *D (B)* will be used to calculate the *HingLoss* move function. We will make the discriminator understand that the result generated by the generator *G(A)* is false by comparing it with the -1 as follows: *HingLoss(D(G(A)),-1)*, and the spectrogram *B* as true since we compare it with the 1, *HingLoss(disc(B),1)*. And then we calculate the average of these two results of the *HingLoss()* function before starting the back propagation step, to modify and regularize the weights.

For the generator we have three parts to calculate in order to realize the back propagation:

- We start by fooling the discriminator by telling it that the result generated by the generator is true using *HingLoss(D(G(A)),1)* when in reality it is false *(GenLoss)*.
- The second part is dedicated to the calculation of the *TravelLoss* function which is one of the three loss functions of the generator.

We start by generating vectors using the Siamese network *S* of the two parts of the spectrogram before the conversion of the generator *(a_1, a_2)*, as well as *G (a_1)* and *G (a_2)* after the generation *G*. All this to check the similarity between the latter which will allow us to preserve the linguistic content.

TravelLoss is: $siamese\ (a_1) - siamese\ (a_2) \ldots X_1$ (3)

And

$siamese\ (G\ (a_1)) - siamese\ (G\ (a_2)) \ldots X_2$ (4)

$$TravelLoss = similarity\ (X_1, X_2) + distance\ (X_1, X_2)^2 \quad (5)$$

With the size of the data that we have until now, we can calculate *MarginLoss* using *siamese (a_1)*, *siamese (a_2)* and sigma which is fixed at 3.

As :

$$MarginLoss = max\ (0, sigma - distance\ (siamese\ (a_1), siamese\ (a_2))) \quad (6)$$

We use this function in the interest of ensuring that the result generated by the Siamese network vector will be sigma different from every other vector generated by this network.

Exemple: $siamese\ (a_1) = siamese\ (a_2) + sigma\ Or\ siamese\ (a_1) - 2*sigma = siamese\ (a_2)$ (7)

Now we have the possibility to evaluate the error cost of the Siamese and do the back propagation to optimize the parameters of the siamese model because we have all the necessary data, *MarginLoss*, *TravelLoss*, gama and beta will be fixed both at 10.

Then:

$$SiameseLoss = (gama * MargingLoss + beta * TravelLoss)/2 \quad (8)$$

In this last step, we noticed that during the training some linguistic data are lost. This is because we adopted the idendity loss in order to preserve the data transformed from *A* to *B* by the generator. This loss preserves linguistic data such as *TravelLoss*. And it also gives another layer of robustness to the generator *G* different from *TravelLoss* because it uses the *B* which is the result we want to reach instead of *A* or *G(A)*.

Such as:

$$IdendityLoss = \| G\ (b1) - b1) \| \quad (9)$$

After having all the results of the necessary constraints like *TravelLoss*, *IdendityLoss* and *GenLoss* we can now calculate the error cost of the generator in order to do the back propagation part to optimize and update the weight of the generator to have better results.

2.4 Experimental results

As the goal of this work is to stop depending on labeled datasets for speech recognition, the training set will contain considerable number of records of an autistic child collected during the use of the system. Which means learning the voice of the child and its own peculiarities by training the generator on its voice before generating an answer with unlabeled dataset.

Thus, we did real time experiments on five autistic children in the autism center, this limited number of children is due to the difficulties of having authorization from the parents and the authority, but we intend to augment it with time. The MelGAN-VC was trained on two hundred examples for each child, in a recurrent way by modifying a training parameter in each iteration, namely the number of examples, the number of epochs and the number of batches. The used optimizers were SGD and Adam.

Experiment 1. After an empirical study, we were able to choose the accurate optimizer and the learning rate for our model, (Table 3).

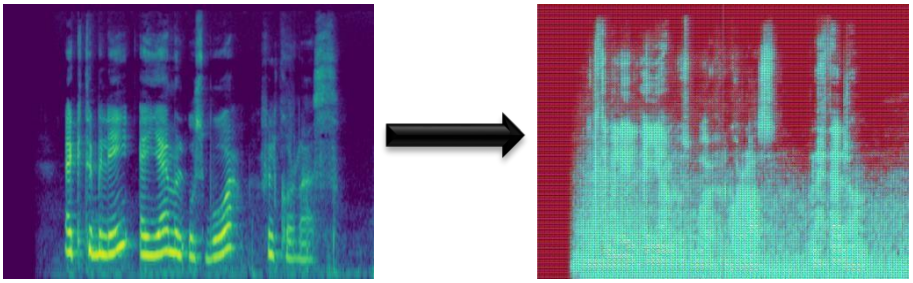
Table 3. The parameters of the transposed convolution.

Model	Optimizer	Learning rate
Generator	Adam	0.0001
Discriminator	Adam	0.0004
Siamese	Adam	0.0001

The training parameters for this experiment are shown in the following table (Table 4):

Table 4. The training parameters.

Training parameters	
Batch	1
Epoch	5
Iteration_lenght	200



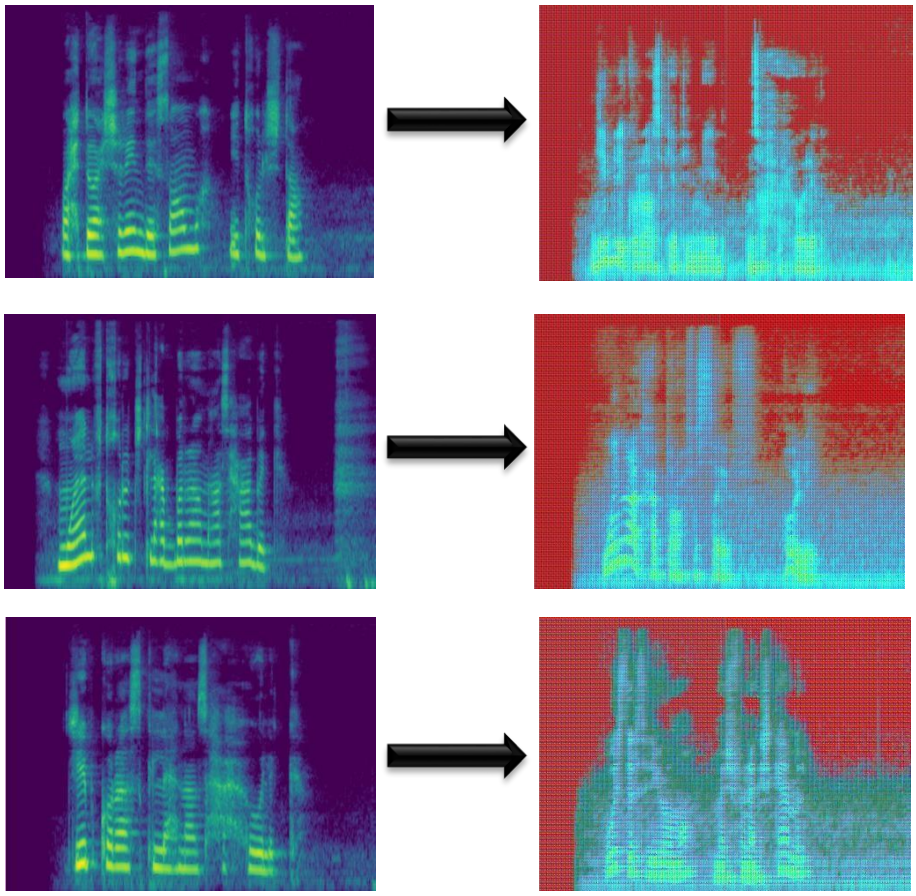


Fig. 10. Original audio (left of each arrow) and generated (right of each arrow) spectrogram samples.

From the obtained results, (see Fig. 10), we chose only the four significant images that show better the resemblance between the generated and the original spectrogram. Based on the human observation of form and color, the system managed to make a remarkable generation.

Experiment 2. The parameters of the transposed convolution for this experiment are shown in the following Table 5:

Table 5. The parameters of the transposed convolution.

Model	Optimizer	Learning rate
Generator	Adam	0.00001
Discriminator	SGD	0.00001
Siamese	Adam	0.00001

The training parameters for this experiment are shown in the following Table 6:

Table 6. The training parameters.

Training parameters	
Batch	1
Epoch	5
Iteration_lenght	200

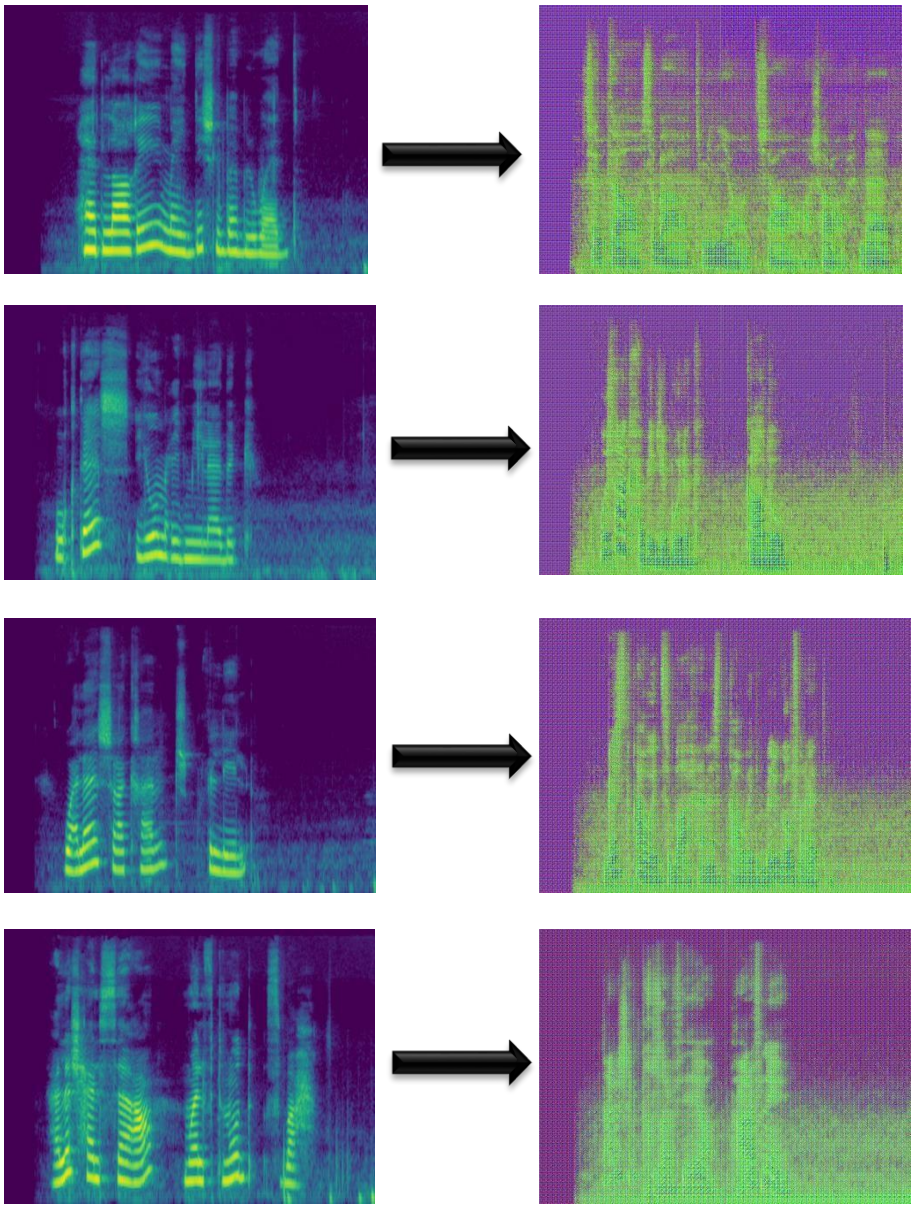


Fig. 11. Original audio (left of each arrow) and generated (right of each arrow) spectrogram samples.

By changing the type of optimizer from Adam to SGD for the discriminator and keeping all other parameters unchanged, it should be mentioned that the system has diverged a bit from the first experiment in terms of colors (see Fig. 11).

Experiment 3. Based on the results of the experiment 2, we decided to keep the same parameters of the transposed convolution and test other training parameters.

The chosen training parameters for this experiment are shown in the following table 7:

Table 7. The training parameters.

Training parameters	
Batch	3
Epoch	3
Iteration_lenght	200

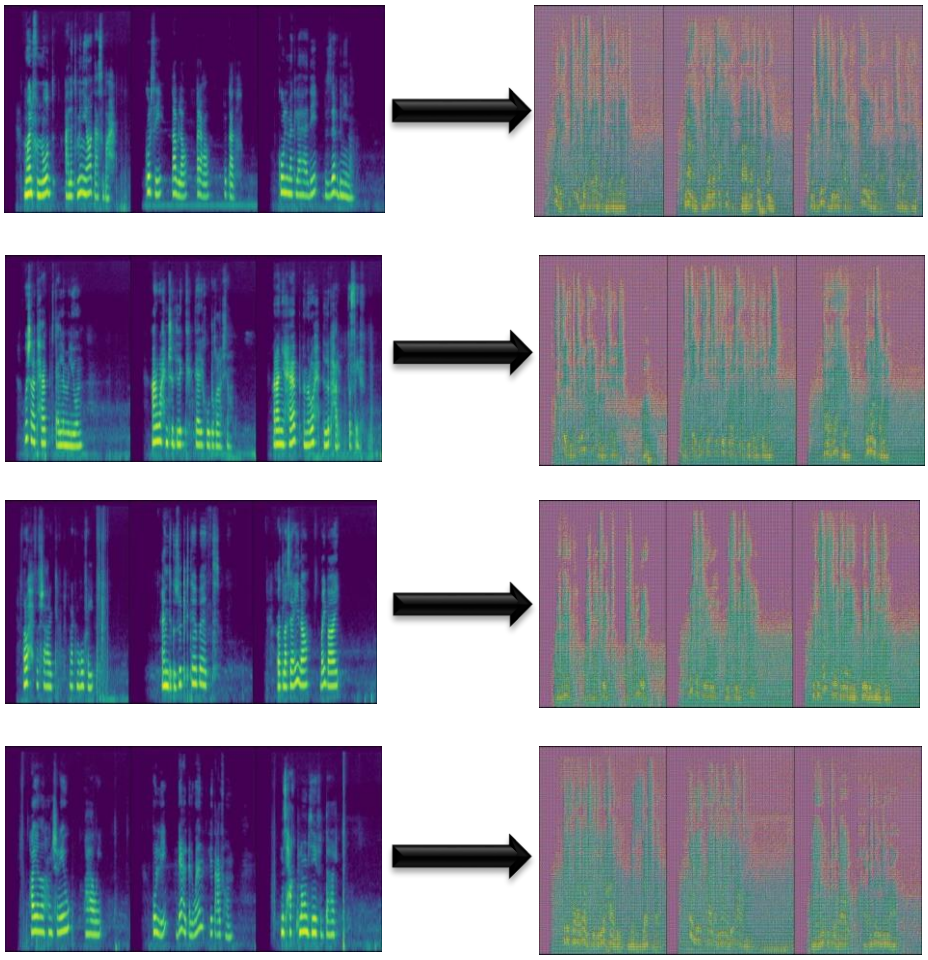


Fig. 12. Original audio (left of each arrow) and generated (right of each arrow) spectrogram samples.

We observe in this experiment that despite the increase of the number of batches from 1 to 3, the system still does not manage to converge. It generates spectrograms with a lot of noise (see Fig. 12).

Experiment 4. The parameters of the transposed convolution for this experiment are shown in the following Table 8:

Table 8. The parameters of the transposed convolution.

Model	Optimizer	Learning rate
Generator	Adam	0.0001
Discriminator	Adam	0.0004
Siamese	Adam	0.0001

The training parameters for this experiment are shown in the following Table 9:

Table 9. The training parameters.

Training parameters	
Batch	3
Epoch	5
Iteration_lenght	200

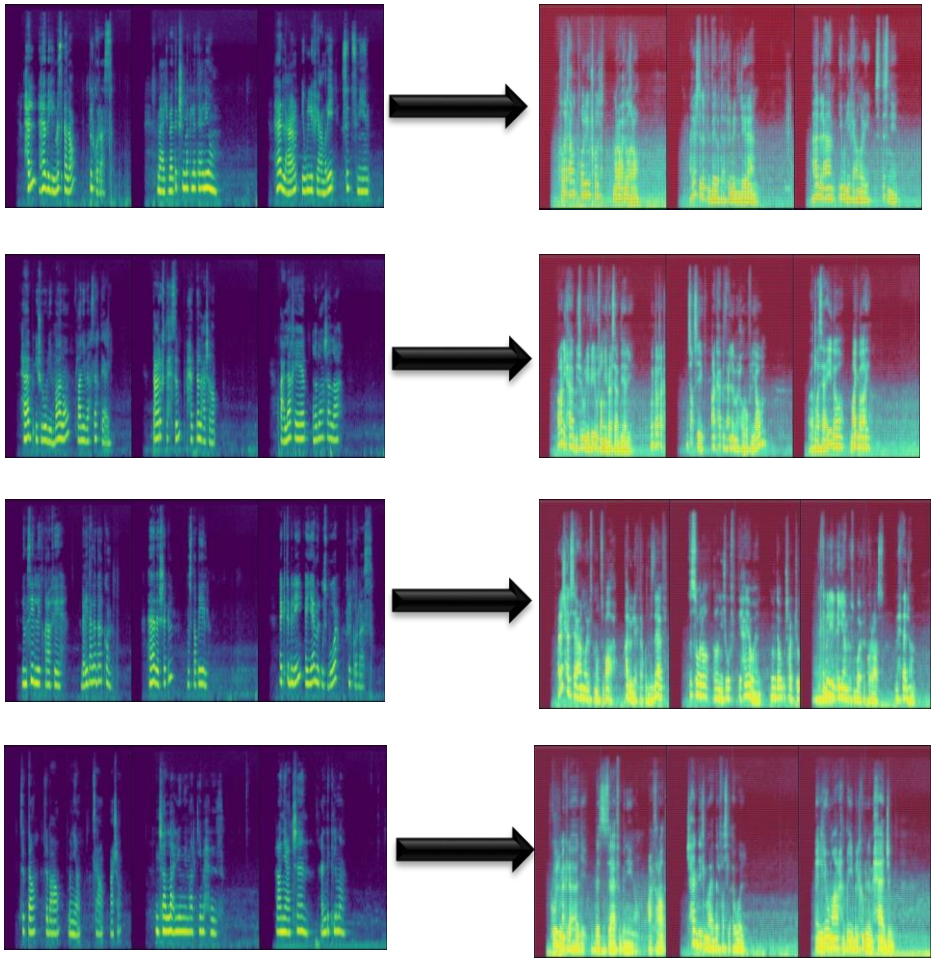


Fig. 13. Original audio (left of each arrow) and generated (right of each arrow) spectrogram samples.

In this last experiment, we can clearly see that the system has managed to converge to the right path. It has generated spectrograms with colors and shapes that are close to those of the input spectrograms, but the only drawback is the presence of noise and this is due to the lack of data (see Fig. 13).

As there is not a specific method to evaluate such results, Passini in [27] has evaluated his generated spectrograms by himself and he mentioned the presence of strong resemblance between the translated music samples and the source ones. Hence, as we work on voices with peculiarities for autistic children, there exists no method to evaluate it, so we depended on the human evaluation and the comparison of the resulted spectrograms with the source ones. Then, the resemblance was strong as it was mentioned before, and we can conclude that this system can be more accurate if we can have bigger dataset to feed our generator.

2.5 Comparison of two audios

For the final phase of evaluating the answer of the child, we propose to compare between the two audios by calculating the similarity rate between the vectorial representations. In addition, by applying the proposed approach explained in section two, we have limited the comparison on only two vectors of features. Therefore, we needed a powerful vectorial representation to avoid using classical features.

Wav2vecs models are speech representations that are trained in an unsupervised way, and they yield remarkable results in speech recognition applications in recent years, even under low-resource conditions [44]. The input speech signal is transformed by the model into its representation, it passes by an encoder network at first to produce the latent representation and later the context network combines time steps in order to obtain the resulted representation.

Wav2Vec 2.0 Model Architecture. The architecture of the wav2vec 2.0 model consists of three main parts (see Fig.14) :

- Convolutional layers that process the raw waveform input to get latent representation.
- Transformer layers, creating contextualized representation.
- Linear projection to output.

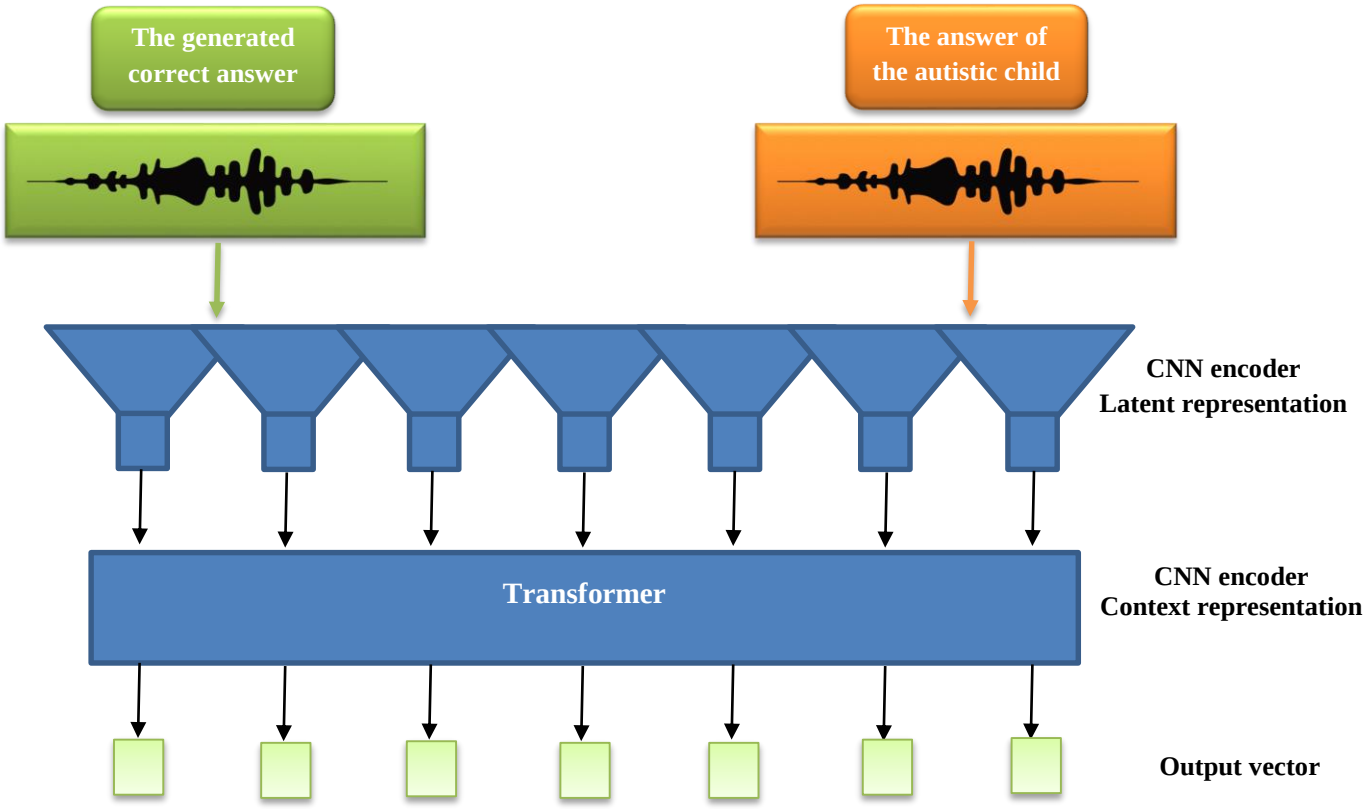


Fig. 14. Wav to vector transformation using wav2vec 2.0 model.

After the extraction of the features vector by wav2vec 2.0, we applied the most used similarity measures used to calculate the distance between two vectors, which are Euclidean, dice, cosine, jacquard and Jensen–Shannon.

Experiment. In order to evaluate the chosen similarity measures we used a dataset of two hundred wave audios from five different persons while they pronouncing same words and different words. The following tables (Table 10) contain the maximum distance and the minimum obtained by each comparison.

Table 10. The obtained results from the comparison experiment.

		Euclidean	Cosine	Dice	Jacquard	Jensen–Shannon
Same person same word	**Min**	**0.0**	**1.0**	0.45	**1.0**	**1.0**
	Max	**0.0**	**1.0**	0.61	**1.0**	**1.0**
Different person different word	**Min**	1.85	0.50	0.25	0.0	0.42
	Max	5.77	0.93	0.51	0.0	0.85
Different person same word	**Min**	1.94	0.64	0.33	0.0	0.53
	Max	4.65	0.94	0.50	0.0	0.87
Same person different word	**Min**	2.04	0.48	0.22	0.0	0.40
	Max	5.60	0.94	0.53	0.0	0.85

By comparing all the results of all the measures for different cases, it is very noticeable that the measures (Euclidean = 0.0 and cosine = jacquard = Jensen–Shannon = 1.0) remain stable when the audios come from the same person and he is pronouncing the same word. In addition, as the Euclidean distance between two objects that are not points is usually defined as the smallest distance between pairs of points from the two objects. Therefore, with a distance of 0.0, it means that they are so similar. Moreover, for other similarity measures such as cosine, jacquard, and Jensen-shannon are equal to 1.0, meaning they match perfectly.

3 Conclusion

In recent years, technologies have offered considerable help to all sectors. Nowadays, clinicians use AI-healthcare systems to monitor and manage patients. Autistic children are not patients but they categorized as people with disabilities because they have difficulty interacting with their peers, teachers and parents, etc. Most of them enjoy connecting to computers or intelligent systems, which proves that AI-based solutions can be an effective tool to improve the instructing and teaching method of social skills.

The main goal of this research work is achieved, which is to develop a tutoring system consists on teaching and evaluating the autistic children. A novel approach has been proposed to overcome the lack of labeled datasets in any language and especially in Arabic language by using the generative adversarial networks. These latter were trained on unlabeled dataset of five autistic children to generate the correct answer with their voices and their peculiarities in pronunciation. Later, the generated voice was compared to the given answer of the child. This comparison was done by converting each wav into a features vector by using one of the most used vectorial representation in the automatic speech recognition field which is wav2vec 2.0. The use of the proposed system shows promising results in teaching individuals with autism in different ways, and it can be used to improve a variety of skills.

As a continuation of this work, we plan to increase our used datasets and add more activities to develop a comprehensive tutoring system for all levels of autism. It also encourages us to expand the work and test the proposed approach on other languages. Generating an avatar that can react with children can also be a brilliant idea for our future work.

Conflict of interest

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Data availability

Data sharing not applicable to this article as we need a permission from the all the children parents, the autism center and the government to publish such data. Thus, we decided to protect the privacy of these children.

References

1. Pennington, M. L., Cullinan, D., Southern, L. B.: Defining autism: Variability in state education agency definitions of and evaluations for autism spectrum disorders. Autism research and treatment (2014).
2. Kim, E. S., Berkovits, L. D., Bernier, E. P., Leyzberg, D., Shic, F., Paul, R., Scassellati, B.: Social robots as embedded reinforcers of social behavior in children with autism. Journal of autism and developmental disorders, 43(5), 1038–1049 (2013).
3. Park, H. R., Lee, J. M., Moon, H. E., Lee, D. S., Kim, B.-N., Kim, J., Kim, D. G., Paek, S. H.: A short review on the current understanding of autism spectrum disorders. Experimental neurobiology, 25(1), 1–13 (2016).
4. Wilens, T. E., Spencer, T. J.: Understanding attention-deficit/hyperactivity disorder from childhood to adulthood. Postgraduate medicine, 122(5), 97–109 (2010).
5. Ghosh, T., Banna, H. Al, Rahman, S., Kaiser, M. S., Mahmud, M., Hosen, A. S. M. S., Hwan, G.: Artificial intelligence and internet of things in screening and management of autism spectrum disorder. Sustainable Cities and Society, 74(June), 103189 (2021). https://doi.org/10.1016/j.scs.2021.103189
6. McCarthy, J.: Artificial intelligence, logic and formalizing common sense. Philosophical logic and artificial intelligence (pp. 161–190). Springer (1989).
7. Quinlan, J. R.: C4. 5: Programs for machine learning. Elsevier (2014).
8. Ashton, K., et al.: That 'internet of things' thing. RFID journal, 22(7), 97–114 (2009).
9. Knight, V., McKissick, B. R., Saunders, A.: A review of technology-based interventions to teach academic skills to students with autism spectrum disorder. Journal of autism and developmental disorders, 43(11), 2628–2648 (2013).
10. Kaur, N., Kaur, A., Dhiman, N., Sharma, A., Rana, R. A.: systematic analysis of detection of autism spectrum disorder: Iot perspective. International Journal of Innovative Science and Modern Engineering (IJISME), 6 (2020).
11. Hyde, K. K., Novack, M. N., LaHaye, N., Parlett-Pelleriti, C., Anden, R., Dixon, D. R., Linstead, E. Applications of supervised machine learning in autism spectrum disorder research: A review. Review Journal of Autism and Developmental Disorders, 6 (2), 128–146 (2019).
12. Jaliaawala, M. S., Khan, R. A.: Can autism be catered with artificial intelligence-assisted intervention technology? a comprehensive survey. Artificial intelligence review, 53(2), 1039–1069 (2020).
13. Moon, S. J., Hwang, J., Hill, H. S., Kervin, R., Birtwell, K. B., Torous, J., Kim, J. W.: Mobile device applications and treatment of autism spectrum disorder: A systematic review and meta-analysis of effectiveness. Archives of disease in childhood, 105(5), 458–462 (2020).
14. Jouaiti, M., Henaff, P.: Robot-based motor rehabilitation in autism: A systematic review. International journal of social robotics, 11(5), 753–764 (2019).
15. Abirami, M., Banu, A. S., Miranda, T. B., Dhivya, M.: A systematic review for assisting the echolalia attacked autism people using robot and android application. International journal of computer applications, 115(6) (2015).
16. Lorenzo, G., Lledó, A., Arráez-Vera, G., Lorenzo-Lledó, A.: The application of immersive virtual reality for students with asd: A review between 1990–2017. Education and Information Technologies, 24 (2018). https://doi.org/10.1007/s10639-018- 9766-7
17. Ha, M. N.: A review of serious game for autism children. Computer Games, Multimedia and Allied Technology (CGAT 2012), 90 (2012).
18. Park, D. S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E. D., Le. Q. V.: Specaugment: A simple data augmentation method for automatic speech recognition. In Proc. of Interspeech (2019).
19. Synnaeve G., et al.: End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures(2019). arXiv, abs/1911.08460
20. Han W., et al.: Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. arXiv (2020).
21. Gulati, A., Qin, J., Chiu, C.C., Parmar, N., Zhang, Y.: Conformer: Convolutionaugmented transformer for speech recognition. arXiv (2020).
22. Lewis, M. P., Simon, G. F., Fennig, C. D.: Ethnologue: Languages of the world, nineteenth edition. Online version (2016). http://www.ethnologue.com
23. Liu, A. H., Lee, H.-Y., Lee, L.-S.: Adversarial training of end-to-end speech recognition using a criticizing language model. arXiv (2018).

24. Baskar, M. K., Watanabe, S., Astudillo, R., Hori, T., Burget, L., Cernocký, J.: Semi-supervised sequence-to-sequence asr using unpaired speech and text. arXiv (2019).
25. Hsu, W.-N., Lee, A., Synnaeve, G., Hannun, A.: Semi-supervised speech recognition via local prior matching. arXiv (2020).
26. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 2672–2680 (2014).
27. Pasini, M.: MelGAN-VC : Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms arXiv : 1910 . 03713v2 [eess . AS] (2019).
28. Kain, A., Macon., M. W.: Spectral voice conversion for text-to-speech synthesis. In Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing No.98CH36181. ICASSP '98 (Cat, volume 1, pages 285–288 vol.1) (1998).
29. Kain, A. B., Hosom, J-P., Niu, X., Santen, J. P. V., Fried-Oken, M., Staehely, J.: Improving the intelligibility of dysarthric speech. Speech communication, 49(9):743–759 (2007).
30. Nakamura, K., Toda, T., Saruwatari, H., Shikano, K.: Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech. Speech Communication, 54(1):134–146 (2012).
31. Inanoglu, Z., Young, S.: Data-driven emotion conversion in spoken english. Speech Communication, 51(3):268–283 (2009).
32. Turk, O., Schroder, M.: Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques. IEEE Transactions on Audio, Speech, and Language Processing, 18(5):965–973 (2010).
33. Toda, T., Nakagiri, M., Shikano, K.: Statistical voice conversion techniques for body-conducted unvoiced speech enhancement. IEEE Transactions on Audio, Speech, and Language Processing, 20(9):2505–2517 (2012).
34. Baer, D.M., Wolf, M.M., Risley, T.R.: Some current dimensions of applied behavior analysis. Journal of Applied Behavior Analysis. 1968;1:91–97. doi: 10.1901/jaba.1968.1-91 (1968).
35. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, 1–8 (2015).
36. Szegedy, C., Com, S. G.: Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift, 37 (2015).
37. Cornec, K. Le. : Apprentissage Few Shot et méthode d ' élagage pour la détection d ' émotions sur bases de données restreintes To cite this version : HAL Id : tel-03143123 Apprentissage Few Shot et Méthode d'Élagage pour la Détection d'Émotions sur Bases de Données Restreintes (2021).
38. Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F.: Recent Progress on Generative Adversarial Networks (GANs): A Survey. IEEE Access, PP(c), 1 (2019). https://doi.org/10.1109/ACCESS.2019.2905015
39. Saxena, D., Cao, J. Generative Adversarial Networks (GANs): Challenges , Solutions , and Future Directions, 54(3) (2021).
40. Kumar, K., Gestin, L., & Courville, A. : MelGAN : Generative Adversarial Networks for Conditional Waveform Synthesis, NeurIPS (2019).
41. Lanham, M.: Generating a New Reality. Book (2021).
42. Wang, D., Dong, L., Wang, R., Yan, D.: Fast speech adversarial example generation for keyword spotting system with conditional GAN Computer Communications, 179(202003), 145–156 (2021). https://doi.org/10.1016/j.comcom.2021.08.010
43. Embarcadero-ruiz, D., Gómez-adorno, H., Embarcadero-ruiz, A., Sierra, G.: Graph-Based Siamese Network for Authorship Verification, 1–24 (2022).
44. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in Neural Information Processing Systems, 33 (2020).