8th ISCA Workshop on Speech and Language Technology in Education (SLaTE 2019)
20-21 September 2019, Graz, Austria

Transfer Learning based Audio Classification for a noisy and speechless recordings detection task, in a classroom context

Mohamed El Hajji[1], *Morgane Daniel*[1], *Lucile Gelin*[1,2]

[1]Lalilo,France
[2]IRIT, Paul Sabatier University, CNRS, Toulouse, France
mohamed@lalilo.com, morgane@lalilo.com, lucile.gelin@irit.fr

Abstract

In this work, we study the effect of Transfer Learning on an audio classification task. The recordings to classify are young children reading aloud isolated words in a classroom context. We aim at detecting which recordings are noisy and/or speechless. We explored both Recurrent Neural Network and Fully Connected Neural Network architectures. To train our classifiers, a Transfer Learning based feature extraction approach is introduced, using the VGGish pre-trained model made available by Google as a feature extractor. Due to pedagogical constraints, the different possible misclassifications do not have the same consequences. Therefore, an alternative metric to the F1 score is presented, which takes into account the pedagogical consequences of the possible misclassifications.

Results show that networks trained on Transfer Learning based features perform better than networks trained on Mel-Frequency Cepstral Coefficients, which are typically used in speech recognition tasks, with a relative improvement of 25% in our metric between the best performing models. These networks also provide a reduction of three to five times of computation time for training.

Index Terms: transfer learning, feature extraction, VGGish, recurrent neural network

1. Introduction

Recently, the optimization of Convolutional Neural Networks (CNN) training, as well as the availability of large datasets such as ImageNet [1], have significantly improved the performance of image classification tasks using networks with increasingly deep architectures like Inception [2], Resnet [3], Alexnet [4] or VGG [5].

Audio classification tasks, also called acoustic event detection, generally use features such as Mel-Frequency Cepstral Coefficients (MFCC), and models based on Gaussian Mixture Models, Hidden Markov Models, or Support Vector Machines [6,7,8]. With the democratization of Deep Learning, CNN [9] and Recurrent neural networks (RNN) [10] are used as well with relatively small datasets such as ActivityNet [11] or TRECvid [12]. With the creation of bigger audio datasets, work has been done to train deep CNN with architectures similar to the ones in image classification, but to classify audio snippets [14].

Transfer Learning (TL) [16] is a method that involves taking a pre-trained source network on a very large dataset, and transferring the knowledge acquired from it to train a new network, adapted to a smaller quantity of data. The source and target tasks can be different : the approach assumes that, being trained on a very large dataset, the network creates in its first layers a sufficiently general data representation to be able to apply it to other tasks on the same type of data. The first layers learn general characteristics of the data while the deeper layers, closer to output, learn specific features of the problem [15].

Lalilo develops a reading assistant for children between the ages of 5 and 7, for both English and French languages. The goal is to adapt the learning of reading to the difficulties of each child. Children are encouraged to read words or sentences out loud, in their native language, and a speech recognition system is used to assess pronunciation and fluency to provide feedback to the child.

Most of the speech recognition systems for children have been trained on corpora containing audio recorded in good conditions, usually with no or little background noise and no overlapping speech. However, Lalilo's assistant is always used in a classroom context with variable and unpredictable babble noise, like overlapping speech from the teacher or other students than the targeted one, and with a lot of different microphones often of low quality. Thus, the collected recordings are likely to contain strong background noise, particularly babble noise, which will deteriorate the accuracy of the speech recognition system. The assistant might then give the child inadequate feedback according to its performances, which can cause frustration. A second problem arises when the child is distracted or has not understood the exercise, leading to a speechless recording.

This paper presents our work on a system of detection of recordings that are noisy and/or do not contain speech of the target child, in order to give an immediate feedback about the recording environment. The rejection of these recordings will avoid misleading feedback given by the speech recognition system due to poor performance in noisy environments, which can frustrate the child who may not understand why the feedback is bad. For this task, we first train a simple fully connected Deep Neural Network (DNN), then a RNN, both fed with MFCCs, used as a baseline to compare results. Then, we use a Transfer Learning method with a pre-trained deep CNN made available by Google, trained on the AudioSet dataset [13]. This network serves as a feature extractor for audio data, features that are then fed to our models in place of the MFCCs in baseline models. In the following, we will first present the dataset on which our experiments have been conducted, then our experimental setup. Finally, we will present and discuss the results achieved by our different models.

2. Dataset

The dataset is an in-house corpus containing 20,000 recordings of 5-7 year-old children reading aloud isolated words. Every recording corresponds to one attempt of a child to read one word aloud and can last from 1 to 10 seconds. 6,000 records are in French and 14,000 in English. These recordings were collected either manually in classrooms, and in this case the recording conditions are relatively good, either via the Lalilo platform.

10.21437/SLaTE.2019-21

The data collected from the platform are not as qualitative as the first ones, mainly because the recorded students are not supervised by their teacher, the ambient noise level can not be controlled and the microphones used in schools are usually of poor quality (like the open, built-in microphones from tablets or laptops). The recordings are then labeled with a cross annotation by two human judges to ensure the quality of the data.

The dataset is divided into 2 classes: *Accepted* and *Rejected*. The former corresponds to good quality recordings, which will be sent to the speech recognition system to be processed, while the latter regroups the noisy and speechless recordings, which we want to reject. On those recordings, the accuracy of the speech recognition would be deteriorated and could lead to a wrong, thus frustrating feedback for the student. The *Rejected* class can be decomposed in 2 sub-classes: *Noise* and *Speechless*, which respectively represent the highly noisy recordings and the recordings that do not contain the voice of the target child. 2000 recordings of the dataset are put apart and constitute the test set, while the rest serve as training/validation data. Both training and test sets have the same class distribution, which can be found in Table 1.

Table 1: *Number of recordings and class distribution*

Set	**Train**		**Test**		**Proportion**
Language	EN	FR	EN	FR	(%)
Accepted	4213	2419	468	268	36
Rejected	8387	2981	932	331	64
Noise	2040	1149	227	128	18
Speechless	6347	1832	705	203	46

3. Experimental Setup

3.1. Feature extraction

3.1.1. Baseline models

For the baseline models, we extract 13 MFCCs, as well as their first and second derivatives, on frames of duration 25 ms with a 10 ms shift. Features are then normalized with a standard scale and used as train data.

3.1.2. Transfer Learning models

We perform Transfer Learning with, as pre-trained model, a deep CNN made available by Google, trained on the AudioSet Dataset [13], which is a database of 6000 hours of Youtube video belonging to 567 classes. called VGGish [14]. The architecture used is the same as the architecture A in [5]. In order to adapt it to our specific task, we remove the last 4 layers to keep only the convolution layers, and freeze the weights of the remaining layers. For every 960 ms of audio data, we generate spectrograms with a window size of 25 ms, a 10 ms shift, and a feature dimension of 64. Thus, for every 960 ms of audio data, we get a 96×64 spectrogram. This data is conveyed to the pre-trained VGGish model, which outputs a 128-dimensional representation vector for every 960 ms of audio. Finally, those vectors are used as features instead of the MFCCs to train our classifiers. We can not compare our models with a baseline model trained on the Audioset dataset without TL because the dataset is not labeled for our specific task.

3.2. Network structure and Hyper-parameters optimization

3.2.1. Classification tasks

We train both binary and multiclass classifications, through several experiments that are presented in Table 2, along with the labels assigned to each class. The last two binary classifications are used together: the probabilities predicted by each of the two networks are averaged to give the final class probability. We will call this classification task *double binary* in the following.

Table 2: *Class labels for each classification task*

File category **Classification type**	**Accepted**	**Noise**	**Speechless**
Multiclass	2	1	0
Binary	1	0	0
Binary Noise	1	0	not used
Binary Speechless	1	not used	0

3.2.2. Network architecture and Hyper-parameters optimization

A variant of the grid search algorithm enables us to choose the best architecture : it varies only two parameters at a time, to find the structure that provides the best results for each classification task and each network type (DNN and RNN).

The results of this grid search allowed us to set the value of all parameters, except the number of neurons per layer and the number of hidden layers, for which we did not find any architecture that is superior to the others for all our tasks. For all our models, we use an Adam optimizer, a ReLU activation function, and the learning and dropout rates are fixed respectively to 0.001 and 0.3.

The loss functions are the binary crossentropy for the binary models, and the categorical-crossentropy for the multiclass models. To set the number of neurons i and the number of hidden layers j, we tested all combinations with i in $\{2^n, n \in \mathbb{N} \,|\, 5 \leq n \leq 9\}$ and j in $\{n \in \mathbb{N} \,|\, 2 \leq n \leq 12\}$.

3.2.3. Language discrimination

We tried both training on a single language or on the full dataset (English + French). Results were always better when training on the full dataset, mainly because there is more data. Results in the next section are thus shown only for models trained on the full dataset.

3.3. Evaluation

3.3.1. Multiclass and binary models comparison

First, we want to be able to compare multiclass models to binary models. A multiclass model can be wrong in predicting that a noisy recording is speechless or inversely and that will lower its metrics, while this kind of error does not matter in our problem. If we refer to Table 2, we see that for binary classification, class 1 represents the Accepted recordings and class 0 represents the Rejected recordings. In the multiclass case, class 2 represents the Accepted recordings, and the Rejected ones are represented by classes 1 and 0. Thus, we will regroup class 0 and 1 as the Rejected class and class 2 will correspond to the Accepted class when computing the metrics.

3.3.2. Metrics

The classical metric of the F1 score is not adapted to our problem since the different misclassifications do not have the same consequences : predicting that a recording should be rejected whereas it should not is always frustrating for the child, who will have to do the exercise again. On the other hand, predicting that a recording should be accepted while it should not, on the worst case, lead to misleading feedback given by the speech recognition system, but might also lead to a correct feedback if the noise is not too strong.

Each network outputs one probability per class, and we must set the threshold such that, as soon as the predicted probability (of one of the classes) reaches this limit, the sample belongs to this class. The usual threshold is equal to the number of classes divided by 2, but taking another value can be beneficial if the possible misclassification errors do not have the same consequences, which is the case here.

For every threshold between 0 and 1 with a 0.01 step, we compute the percentage of Rejected recordings Correctly Classified (RCC) and the percentage of Accepted recordings Misclassified (AM). The difference between those two percentages is defined by the function g in Eq. (1), where t is the threshold applied.

$$\forall t \in \big[0:1\big], g(t) = RCC(t) - AM(t) \quad (1)$$

This function represents the performance of the model : higher values of g mean that RCC is much higher than AM, which means the network is effective for our task. As this function is defined on a discrete space, it has a maximum value, associated with a particular threshold value called t_m. At this threshold, the model is at his peak of performance. However to avoid frustrating the children, we need to minimize the AM percentage, i.e. we have to make sure that the value of $AM(t_m)$ is low enough.

We can then define the function f (Eq. (2)) by setting the threshold t_0 so that AM(t_0) is fixed.

$$\forall AM(t_0) \in \big[0:1\big], f(AM(t_0)) = RCC(t_0) \quad (2)$$

We can now define three metrics for different values of AM(t_0):

- $f(0.10)$: the percentage of Rejected recordings Correctly Classified (RCC) with 10 % of Accepted recordings Misclassified (AM)
- $f(0.15)$: the percentage of Rejected recordings Correctly Classified (RCC) with 15 % of Accepted recordings Misclassified (AM)
- $f(AM_m)$: the percentage of Rejected recordings Correctly Classified (RCC) with AM_m = AM(t_m)

3.3.3. Merging models

Finally, for each of our network architecture (DNN, RNN), we experiment merging the best models for each classification task: for a given sample, we apply a decision function on the prediction of the best multiclass, binary and double binary models, in order to output only one probability or prediction.

Thus, the different models can be compensated for, and generally the resulting model is more effective than all of the baseline models. It is also this idea which is implemented during the Double Binary classification. In our case, we use two classical merge functions, which are **Mean**, where we average the probabilities outputs by each network to have the total probability, and **Majority**, where we do a majority vote, that requires an odd number of models.

4. Results and Analysis

4.1. Without Transfer Learning

In Table 3, we present the performances, measured with the previously defined metrics, for baseline models trained with MFCCs. This table is divided into two parts, one for each type of network (DNN and RNN). The last column displays the value $f(AM_m)$ (i.e. the value of RCC at the threshold t_m that maximizes g(t), see Eq. (1)) as well as the AM_m value, which is the value of AM at the threshold t_m. The model that provides the biggest value of $f(AM_m) - AM_m$ (i.e. the max of the function g) is thus considered the best.

Table 3: $f(0.10)$, $f(0.15)$ *and* $f(AM_m)$ *values (%) for baseline models, with* $AM_m = AM(t_m)$

Metric / **Model**	$f(0.10)$	$f(0.15)$	$f(AM_m)/AM_m$
Multiclass (DNN)	40	49	77 / 33
Binary (DNN)	39	46	76 / 34
Double binary (DNN)	43	52	74 / 28
Merged model (DNN)	-	-	-
Mean	42	**56**	**78 / 29**
Majority	**44**	55	80 / 33
Multiclass (RNN)	48	61	77 / 27
Binary (RNN)	46	58	77 / 27
Double binary (RNN)	50	64	**79 / 25**
Merged model (RNN)	-	-	-
Mean	**51**	**65**	77 / 25
Majority	50	**65**	80 / 26

Regarding the DNN models, we can observe that the double binary model is on average 10% more effective than the two other models alone. The merged models are also more effective than any model alone, for all the metrics. Considering that the double binary is a merged model, merged models perform on average 12% better than single models. We can also see that the AM_m values are between 29% and 34%, which is too high for the system to be used in the Lalilo platform.

Another observation is that the RNNs improve the metrics by 20% on average. Even the worst RNN model is more effective than the best merged DNN model. In the same way as for DNNs, the Double binary RNN is performing better, with an average improvement of 5%, than the multiclass or the binary RNN. Merged RNNs are on average 6% more effective than simple RNNs. Finally, we can see that the average value of AM_m dropped from 32 to 26 %, while the value of $f(AM_m)$ increased from 77 to 78%, which means that RNN models are better at correctly classifying Rejected recordings while misclassifying a lower rate of Accepted recordings.

4.2. With Transfer Learning

In Table 4 are displayed the performances, in the same form as Table 3, for the models trained with Transfer Learning.

Table 4: $f(0.10)$, $f(0.15)$ *and* $f(AM_m)$ *values for models with Transfer Learning, with* $AM_m = AM(t_m)$ *(%)*

Metric / **Network**	$f(0.10)$	$f(0.15)$	$f(AM_m)$/AM_m
Multiclass (DNN)	60	71	82 / 22
Binary (DNN)	57	71	82 / 22
Double binary (DNN)	60	70	82 / 22
Merged model (DNN)	-	-	-
Mean	60	**74**	**85 / 24**
Majority	**61**	73	**85 / 24**
Multiclass (RNN)	**65**	74	**80 / 18**
Binary (RNN)	**65**	73	**80 / 18**
Double binary (RNN)	62	73	82 / 21
Merged model (RNN)	-	-	-
Mean	64	**76**	81 / 19
Majority	**65**	74	81 / 19

As expected, the TL-based approach brings a huge improvement in all the metrics with an average relative improvement of 3% for the best DNN model and 27% for the best RNN model on $f(0.10)$ in comparison the best DNN and RNN with MFCC. Regarding the DNNs, all the models alone have slightly the same performances, and we can see that merged models do not bring significant improvement over the models alone : the Mean model is 4% more effective than the best single network. The AM_m values are also lower than without Transfer Learning, with 23 % (-40%, relative) on average, while the $f(AM_m)$ value increased to 83 % (+8%).

For RNNs, we can note a small improvement of 3 to 8% for the best models, depending on the metric, compared to DNN with Transfer Learning. The main improvement is in the value of AM_m, that ranges between 18 and 21 % while for DNNs, the lowest value is 22 %. $f(AM_m)$ value also dropped but the difference between $f(AM_m)$ and AM_m, i.e. the maximum of the function g (Eq. (1)), is still the same. This means that TL-RNN models are performing equally to TL-DNNs at their peak of performance, nonetheless with a lower rate of misclassification of Accepted recordings. We can also see that the double binary model is less effective than the others, and that the merged models are not useful here since the multiclass model performs equally.

5. Discussion

By comparing the first parts of Tables 3 and 4, we can clearly see the interest of Transfer Learning: in average, f(0.1) and f(0.15) are improved by 55%, and $f(AM_m)$ gains 10% while AM_m drops from the range 25-34% to 22-24%. When attempting to reach a reasonable rate of 74% of correctly rejected recordings, the best baseline model rejects 25% of accepted recordings, percentage that is not acceptable in the Lalilo platform, while the best TL model misclassifies only 15% of accepted recordings. This result shows that the information brought by the features from VGGish are more representative of our data than MFCCs.

Without TL (Table 3), RNNs are improving metrics by 15 to 20 % in comparison to DNNs, which means that temporal information brings good insight into the problem. With TL (Table 4), this improvement is only between 3 and 8 %, so the information in VGG features does not seem to be totally additional with temporal information.

We can also observe that between the best RNN without TL (Table 3) and the best DNN with TL (Table 4), the DNN is performing better, with a big improvement (+15% to +20%, depending on the metric). As the DNN trained with VGG features does not use temporal information, VGG features are then more representative than MFCC combined with temporal information (MFCC+RNN), even though previous results show the importance of temporal information in our problem.

We know that merged models are useful when the models cannot build a general representation of the problem. As without Transfer Learning, merged models improves the metrics by 10 to 15 %, while with TL, these models performs equally or worse than models alone, we can conclude that with VGGish features, single models manage to create an accurate representation of our problem, that models without TL cannot produce alone. This result is related to the fact that with MFCC, we have 4000 features per second, while with VGGish, we only have 128 features per second of recording. Thus, a single model is able to take into account all the characteristics of a problem with VGG features, which makes the merging of models unnecessary. Another advantage is that with 30 times less features, models can build a general representation of the problem with a shallower architecture than with MFCCs : from 9 to 12 layers with 512 neurons per layer for the network without TL, networks with VGGish use only 2 to 4 layers with 128 to 256 neurons per layer. Thus, models are also training faster : RNNs trained with TL complete an epoch in 1 minute while those trained with MFCC take 3 to 5 minutes.

A drawback of TL is that if we want to use these networks in the Lalilo platform, we also have to load the VGGish model in order to extract the features. As this network is really deep, the time taken to load the model must be taken into account if we want to use this network in our platform.

6. Conclusion

In this work, we study the impact of Transfer Learning on the performance of neural networks for Acoustic event detection on recordings where young children are reading aloud isolated words in a classroom context. To achieve that, we use the pre-trained VGGish model [14] as a feature extractor. We train Deep Neural Networks and Recurrent Neural Networks with the classical MFCCs as a baseline, and then on the features extracted by the pre-trained VGGish model. We show that networks trained with features from VGGish are performing better than networks trained with MFCCs, while being shallower. The best combination is reached for VGG features with RNN, with a mean relative improvement of 22% compared to same model with MFCC. Finally, even DNNs trained with features from the pre-trained VGGish model are 18% more effective in average than RNNs trained with MFCCs, which shows that features from the Transfer Learning are far more representative of our dataset than the combination of MFCCs and temporal structure of the recording, while using 30 times less features to represent the data.

7. References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei, "Imagenet: A large-scale hierarchical image database," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248255.*.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567, 2015.*

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385, 2015*

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems, 2012, pp. 10971105.*

[5] K Simonyan, A Zisserman, "Very deep convolutional networks for large-scale image recognition" *ICLR (2015)*

[6] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," *Signal Processing Conference, 2010 18th European. IEEE, 2010, pp. 12671271.*

[7] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters, vol. 31, no. 12, pp. 15431551, 2010.*

[8] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," *International Evaluation Workshop on Classification of Events, Activities and Relationships. Springer, 2006, pp. 311322.*

[9] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," *arXiv preprint arXiv:1604.07160, 2016.*

[10] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 6440644*

[11] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961970.*

[12] G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F.Smeaton, G. Quenot, M. Eskevich, R. Aly, and R. Ordelman, "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking," *Proceedings of TRECVID 2016. NIST, USA, 2016.*

[13] Gemmeke, J. et. al " Audio set : an ontology and human-labelled dataset for audio event " *ICASSP 2017*

[14] Hershey, S. et. al " CNN Architectures for Large-Scale Audio Classification " *ICASSP 2017*

[15] Yosinski J, Clune J, Bengio Y, and Lipson H. " How transferable are features in deep neural networks? " *Advances in Neural Information Processing Systems 27 (NIPS 14), NIPS Foundation, 2014.*

[16] S. J. Pan and Q. Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering, 22(10):13451359, 2010*