# Speaker Inconsistency Detection in Tampered Video

Pavel Korshunov
Idiap Research Institute
Martigny, Switzerland
pavel.korshunov@idiap.ch

Sébastien Marcel
Idiap Research Institute
Martigny, Switzerland
sebastien.marcel@idiap.ch

***Abstract*—With the increasing amount of video being consumed by people daily, there is a danger of the rise in maliciously modified video content (i.e., 'fake news') that could be used to damage innocent people or to impose a certain agenda, e.g., meddle in elections. In this paper, we consider audio manipulations in video of a person speaking to the camera. Such manipulation is easy to perform, for instance, one can just replace a part of audio, while it can dramatically change the message and the meaning of the video. With the goal to develop an automated system that can detect these audio-visual speaker inconsistencies, we consider several approaches proposed for lip-syncing and dubbing detection, based on convolutional and recurrent networks and compare them with systems that are based on more traditional classifiers. We evaluated these methods on publicly available databases VidTIMIT, AMI, and GRID, for which we generated sets of tampered data.**

## I. Introduction

Recent advances in automated video and audio editing tools, generative adversarial networks (GANs), and social media allow creation and fast dissemination of high quality tampered video content. Such content already led to appearance of deliberate misinformation, coined 'fake news', which is impacting political landscapes of several countries [1]. A recent surge of videos, often obscene, in which a face can be swapped with someone else's using neural networks, so called 'deepfakes'[1], are of a great public concern[2]. Therefore, the development of effective tools that can automatically detect tampered audio-visual content is of paramount importance.

In this paper, we focus on detecting audio-visual tampering in a video of speaking person, i.e., the inconsistencies between video and audio tracks. We present a set of publicly available databases with disproportionately large and challenging tampering video sets and, using these databases, benchmark several approaches for tampering detection. The problem of speaker audio-visual inconsistencies detection is related to dubbing and lip-syncing detections, therefore, we look into the solutions proposed for those problems. The main difference of tampering is that it has a malicious intent and is meant to spoof the viewer into thinking that it is the original video.

Typically, most of the latest approaches [2], [3], [4], [5], [6], [7] for lip-syncing or dubbing detection focus on extracting separate feature sets for audio and video. For audio, mel-scale frequency cepstral coefficients (MFCC) are usually used, while different visual features, varying from optical flow [3] to features learned with deep neural networks (DNNs) [6], are extracted from the mouth region of a face. The features then undergo some processing before they are fed into a classifier, best performing examples including long short-term memory (LSTM) [4] or convolutional neural networks [7].

In this paper, building upon related work, we selected a range of approaches suitable for audio-visual inconsistency detection and performed a preliminary study of different feature processing techniques, classifiers, and their parameters on different databases with tampering attacks. For our tampering detection systems, we used MFFCs as audio features [3] and distances between mouth landmarks as visual features (inspired by [8]). We explored different ways to post-process the features, including ways to combine two types of features, reduce the dimensionality of blocks of features with principal component analysis (PCA), and project both modalities into a common space with canonical correspondence analysis (CCA). We also considered different classifiers, including Gaussian mixture model (GMM), support vector machine (SVM), multilayer perceptron (MLP), and LSTM.

For the databases, we have selected three different public databases with audio-visual data: VidTIMIT[3], AMI corpus[4], and GRID corpus[5]. For each video in a database, we generated five tampered versions by randomly replacing audio track of the person in the video with an audio from another person. Also, it should be noted that videos in VidTIMIT and GRID databases were shot in controlled environments (people are facing camera and reciting predetermined short phrases), while AMI database consists of informal meeting recordings (profile faces, mouth occlusions, unclear speech, etc.), which means it has a more practical and realistic data.

To allow researchers to verify, reproduce, and extend our work, we provide all implementations of the evaluated systems and scripts for generation of tampered data for the databases as an open source Python package available to public[6].

Therefore, the main contributions of this paper include (i) evaluation of different approaches for audio-visual tampering detection, (ii) three databases with tampering videos, and (iii) open source implementation of evaluated systems and scripts for generation of tampered video.

[1]Open source code: https://github.com/deepfakes/faceswap
[2]BBC (Feb 3, 2018): http://www.bbc.com/news/technology-42912529
[3]http://conradsanderson.id.au/vidtimit/
[4]http://groups.inf.ed.ac.uk/ami/download/
[5]http://spandh.dcs.shef.ac.uk/gridcorpus/
[6]Open source code: https://gitlab.idiap.ch/bob/bob.paper.eusipco2018

ISBN 978-90-827970-1-5 

(a) VidTIMIT (b) AMI (c) GRID

Fig. 1. Example screenshots from three databases used in the experiments.

## II. Databases and protocol

Since there is lack of public databases with tampered videos, we looked into available databases with recordings of people speaking in front of the camera and generated our own sets of tampered videos. We have selected three databases (see Table I), VidTIMIT, as an example of small 'toy' database (10 of 3 sec videos per person), GRID corpus, consisting of a large set of short videos (1000 of 3 sec videos per person) in high definition of people facing camera and speaking very short similar-sounding sentences, and AMI corpus, as the most realistic representation of a practical scenario for tampering detection, since it contains recordings of people in office meetings informally talking to each other.

Since AMI database contains both panoramic and videos of individuals, we considered only 977 close up camera videos of consistent speech by the person in the camera, minimizing the crosstalk. Total length of the genuine set is about 6 hours and average video length is 22 sec.

For each database, tampered videos were generated by taking genuine videos and replacing audio track with randomly selected audio from five other speakers. So, for each video from the genuine set, we generated five tampered versions, where video and audio tracks are mismatched. This creates an imbalance in the dataset with five times more attacks than genuine videos, effectively, simulating a realistic scenario of having many tampered versions of a genuine video.

Both genuine and tampered parts of the databases were split into training (*Train*) and development (*Test*) subsets as shown in Table I. To avoid bias during training and testing, we arranged that the same person would not appear in both sets.

TABLE I
Details for VidTIMIT, AMI, and GRID databases.

| Database | Type of data | Train | Test | Total |
|---|---|---|---|---|
| VidTIMIT | subjects | 22 | 21 | 43 |
| | time (hours) | 0.25 | 0.26 | 0.51 |
| | genuine | 220 | 210 | 430 |
| | tampered | 2 | 995 | 2,033 |
| AMI | subjects | 42 | 36 | 54 |
| | time (hours) | 3.82 | 2.28 | 6.1 |
| | genuine | 613 | 364 | 977 |
| | tampered | 2,732 | 1,934 | 4,666 |
| GRID | subjects | 17 | 16 | 33 |
| | time (hours) | 14.01 | 13.19 | 27.2 |
| | genuine | 17,000 | 15,890 | 32,891 |
| | tampered | 79,479 | 75,646 | 155,125 |

### A. Evaluation protocol

Using the computed scores, false acceptance rate (FAR) and false reject rate (FRR) are computed for each possible threshold $\theta$:

$$\begin{aligned} \mathrm{FAR}(\theta) &= \frac{|\{h_{neg} \mid h_{neg} \geq \theta\}|}{|\{h_{neg}\}|} \\ \mathrm{FRR}(\theta) &= \frac{|\{h_{pos} \mid h_{pos} < \theta\}|}{|\{h_{pos}\}|}, \end{aligned} \tag{1}$$

where $h_{pos}$ is a score for genuine samples and $h_{neg}$ is a score for the tampered samples. Threshold $\theta$, at which these FAR and FRR are equal leads to an equal error rate (EER), which is commonly used as the single value metric of the system performance.

## III. Features and classifiers

The goal of the considered tampering detection system is to distinguish genuine video, where lip movement and speech are synchronized, from tampered video, where lip movements and audio, which may not necessarily be speech, are not synchronized. The stages of such system include feature extraction from video and audio modalities, processing these features, and then, a two-class classifier trained to separate tampered videos from genuine.

### A. Video features

To extract visual features that characterize lip movements, we need to reliably detect mouth region in the video. For that, we use OpenPose[7] for body pose [9] and face landmarks

[7]https://github.com/CMU-Perceptual-Computing-Lab/openpose

ISBN 978-90-827970-1-5 © EURASIP 2018

(similar to hands detection in [10]) detection. Based on these detections, we estimate whether the face is frontal (e.g., AMI database contains a lot of profile faces) and then, to characterize the mouth movements, we compute 42 different distances between 20 detected points of the mouth (see Figure 3 for an example), i.e., using points 48 to 64 in Figure 2. These features, though simple, are quite effective, as it is shown in [8], where mouth landmarks were used to generate realistic mouth movements from a given audio speech. In this paper, because mouth features are significantly different in profile faces, we consider frontal faces only.

### B. Audio features

As per the latest related work [3], [4], [5], [6], we also use 13 MFCC features with their delta, double-delta derivatives [11], and energy (40 coefficients in total) to characterize speech in audio. MFCCs are computed from a power spectrum (power of magnitude of 512-sized FFT) on 20ms-long windows with 10ms overlap. Prior to the computation, we pre-emphasize with 0.97 coefficient, apply Hamming window, and normalize raw speech signal by subtracting its mean.

### C. Processing features

There are two main ways to incorporate video and audio modalities into the final system: (i) merge the outputs of two neural networks (one for each modality), i.e., by using contrastive loss as in [4], [7] for detecting out of sync audio shifts, or (ii) combine two types of features prior to the classification, as most of other works use. We follow the second approach, as it is suitable for different types of problems, including tampering detection.

Therefore, we concatenate visual features and MFCC features into one joint vector. Since visual features are extracted at 25 per seconds (as per the video frame rate) and audio features at 100 per seconds (MFFCs are computed on windows shifted by 10ms), we tried the following ways to combine features: (i) upsample video frame rate to match 100 fps of the audio, (ii) downsample audio features to match video 25 fps [5], and (iii) for a fixed temporal window, combine all features into one vector [3]. Third option allows preserving local temporal context in each resulted feature vector that can be learnt by the classifier. Also, after extensive experiments, this approach led to better and more stable results for different databases. In our experiments, we varied the size of the temporal block from 0.08 sec to 0.4 that lead to the joint feature vector varying from 402 to 2020 values.

Principal component analysis (PCA) is often used to reduce the dimensionality of the feature vector. We adopted this approach to reduce the size of the joint visual-audio feature vector, and used 60 (about 95% of variance depending on the database), 80, and 100 (about 98% variance depending on the database) components in the experiments. For each database, the Train set was used to train PCA matrix, which was then applied to all combined features for all samples in the database.

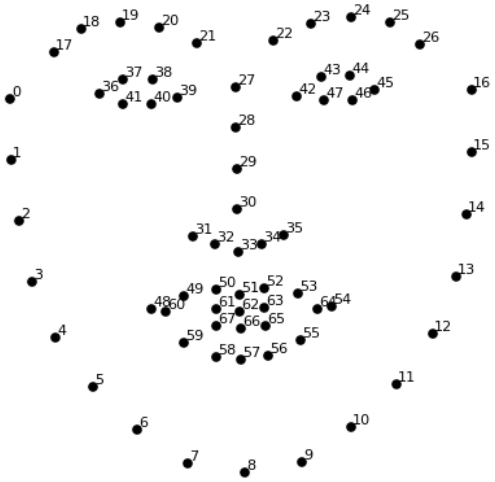


Fig. 2. The 68 landmarks detected by dlib library. This image was created by Brandon Amos of CMU who works on OpenFace.

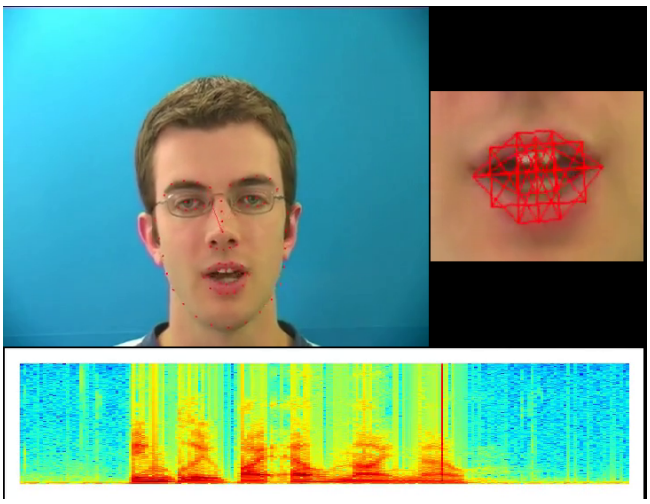


Fig. 3. Screenshot from tampered video (GRID corpus) showing detected visual features and spectrogram.

Canonical-correlation analysis (CCA) is also sometimes [2], [3] used to harmonize features of two modalities prior to the dimensionality reduction, but, in our experiments, we found this technique to have little effect on the results (about 1% reduction in error) and, therefore, do not report it in this paper.

### D. Classifiers

In our experiments, we used Gaussian mixture model (GMM), with varying number of components from 8 to 32, support vector machine (SVM), with best parameters estimated by cross-validation on smaller subsets, multilayer perceptron (MLP), with varying hidden layer size from 8 to 64, and long short-term memory (LSTM) classifiers, with varying cell size from 8 to 64.

The computed features (see Figure 3 for an illustration) from the Train sets of the databases were used to train the models of the classifiers. The EER (see Section II-A for details) values

ISBN 978-90-827970-1-5 © EURASIP 2018

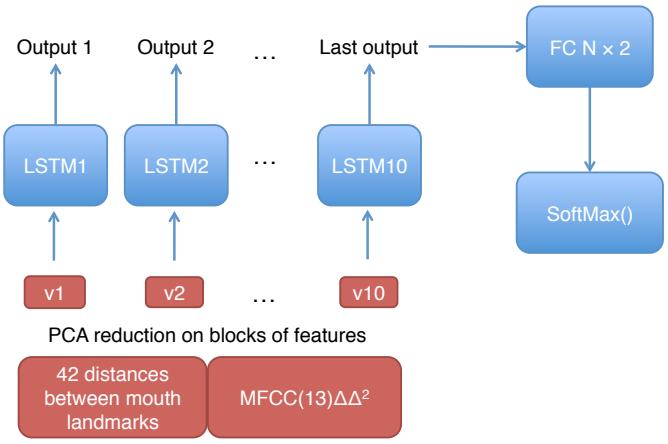



Fig. 4. Features and the architecture of LSTM network.

were computed for both Train and Test sets separately, so we can quickly evaluate if the model overfitted on the training set. For GMM, only one model was trained for genuine data and the mean log-likelihood value determined whether the evaluated sample is genuine (value is lower than EER threshold) or tampered (value is higher than EER threshold).

### *E. LSTM-based system*

The architecture of LSTM-based system is presented in Figure 4. We used one-layer LSTM network implemented with Tensorflow[8] with sigmoid activation function used in the inner cells. The output of the last LSTM cell is forwarded the fully connected layer with two neurons corresponding to the two output classes (genuine and tampered). A SoftMax cross entropy loss function is computed and Adam optimization algorithm [12] with constant learning rate 0.001 is used to optimize the loss.

Since LSTM network learns a temporal context, the PCA-reduced multimodal features are combined into sliding windows to form an input to LSTM-based system. The size of the temporal window is also a parameter of the system.

Some of the important implementation details of the LSTM-bases systems are as follows:

- The training data is balanced in terms of the number of samples from genuine and tampered classes. Since in our databases, genuine set is much smaller than tampered, the data from tampered set is randomly sampled to produce a similar number of features as in the genuine set;
- LSTM is trained using random mini batches of size 16 of sliding windows;
- To improve convergence of LSTM, the data from different classes is fed into the network in a random order to increase the chance that each mini batch has a mixture of samples from different classes.

[8]https://www.tensorflow.org

TABLE II
EER VALUES OF LSTM-BASED TAMPERING DETECTION SYSTEM FOR TRAIN AND TEST SETS FROM THREE DATABASES.

| Database | System | Train (%) | Test (%) |
|---|---|---|---|
| VidTIMIT | **LSTM(64), blk5, pca60, win10** | **0.56** | **24.74** |
| | LSTM(32), blk5, pca60, win10 | 1.26 | 24.74 |
| | MLP (64), blk5, pca60, win10 | 9.03 | 53.45 |
| | SVM, blk2, pca60 | 27.36 | 56.18 |
| | SVM, blk5, pca60 | 3.65 | 30.36 |
| | SVM, blk5, pca100 | 0.00 | 45.28 |
| | GMM32, blk5, pca60 | 2.62 | 56.09 |
| AMI | **LSTM(64), blk5, pca60, win10** | **1.55** | **33.86** |
| | LSTM(32), blk5, pca60, win10 | 1.47 | 34.68 |
| | MLP (64), blk5, pca60, win10 | 24.48 | 41.21 |
| | SVM, blk2, pca60 | 16.80 | 48.39 |
| | SVM, blk5, pca60 | 24.35 | 50.06 |
| | SVM, blk5, pca100 | 23.15 | 54.68 |
| | GMM32, blk5, pca60 | 50.41 | 47.84 |
| GRID | **LSTM(64), blk5, pca60, win10** | **0.73** | **14.12** |
| | LSTM(32), blk5, pca60, win10 | 0.93 | 15.25 |
| | MLP (64), blk5, pca60, win10 | 7.38 | 28.58 |
| | SVM, blk2, pca60 | 4.68 | 30.06 |
| | SVM, blk5, pca60 | 6.54 | 23.93 |
| | SVM, blk5, pca100 | 13.92 | 35.06 |
| | GMM32, blk5, pca60 | 41.33 | 46.81 |

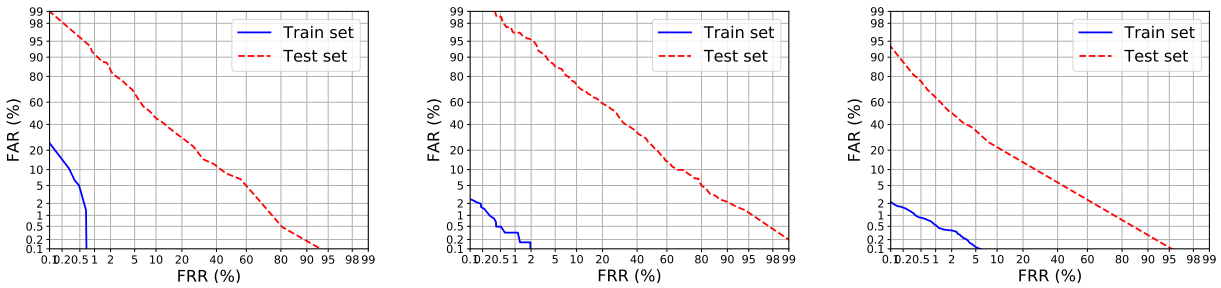
## IV. EVALUATION RESULTS

Different parameters for feature processing and classifiers, coupled with experiments on three databases: VidTIMIT, AMI, and GRID, led to more than 300 experiments, which can be reproduced with the provided open source code[6].

The selected results on three databases are shown in Table II, which presents EER for different systems computed on both Train and Test sets of the databases. In the 'System' column, the system with its main parameter is indicated (e.g., cell size for the LSTM or number of components for GMM), followed by the size of the feature block used (see Section III-C), followed by PCA with the number of components, and, for LSTM systems, the length of the temporal window. It can be noted from the table that LSTM-based system shows the best performance for all databases. In fact, it is the only system that could consistently perform on all databases, while other systems show EER close to 50% on Test set. However, the fact that EER value for LSTM-based system is below 1% for the Train set but higher than 14% for the Test set, means the system overfit on the training data. To illustrate the performance of the best LSTM-based tampering detection system in more details, we plot detection error tradeoff (DET) curves for different databases in Figure 5.

By looking at Table I with database sizes and the results in Table II, we can note that the LSTM-bases system has the lowest EER for Test set on GRID database, while it is the largest database with more than 14 hours of genuine and 70 hours of tampered training data. Subjects in GRID database speak very similar short sentences, leading to speech in tampered video to become similar to genuine, as illustrated by the Figure 6. Although, VidTIMIT also contains frontal faces and short sentences, there are only 10 video per person instead of a 1000 in GRID. Hence, LSTM performs better on

ISBN 978-90-827970-1-5 © EURASIP 2018




(a) VidTIMIT (b) AMI (c) GRID

Fig. 5. DET curves for LSTM-based tampering detection system on different databases.

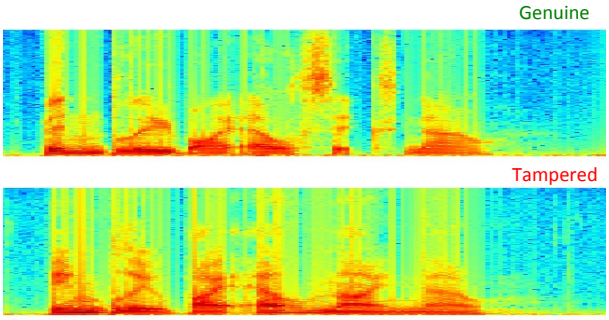



Fig. 6. Similar looking spectrograms of genuine and tampered speech from GRID corpus.

GRID due to the lack of training data in VidTIMIT and AMI. AMI data is also more challenging with multiple occlusions of mouth region, profile faces, and crosstalk from other speakers.

## V. Conclusions

In this paper, based on the related work on lip-syncing and dubbing detection, we benchmarked systems for tampering detection, specifically, audio-visual inconsistencies in videos of speaking people. By using deltas between mouth landmarks for visual features and MFCCs for audio, we considered several feature processing methods with different parameters and different classifiers, including Gaussian mixture model (GMM), support vector machine (SVM), multilayer perceptron (MLP), and long short-term memory (LSTM) networks. The evaluations were done on three publicly available databases, namely, VidTIMIT, AMI, and GRID, which we augmented with tampered video data. The experiments demonstrate that only LSTM-based system can consistently detect tampered data in three databases, though, more efforts need to be put into improving the accuracy of such systems.

In the future, more complex architectures of neural networks, including LSTMs and 3D convolutional networks, should be explored and larger databases are necessary for training such systems. Also, with appearance of videos, where faces are automatically and unnoticeably swapped using generative adversarial networks (GANs), the tampering detection systems need to be trained and tested on such video data.

## Acknowledgment

This research was sponsored by the United States Air Force, Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-16-C-0170. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of AFRL or DARPA.

## References

[1] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.

[2] M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp, "Audiovisual synchronization and fusion using canonical correlation analysis," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1396–1403, Nov 2007.

[3] N. Le and J.-M. Odobez, "Learning multimodal temporal representation for dubbing detection in broadcast media," in *Proceedings of the 2016 ACM on Multimedia Conference*, ser. MM '16. New York, NY, USA: ACM, 2016, pp. 202–206. [Online]. Available: http://doi.acm.org/10.1145/2964284.2967211

[4] J. S. Chung, A. W. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," *CoRR*, vol. abs/1611.05358, 2016. [Online]. Available: http://arxiv.org/abs/1611.05358

[5] E. Boutellaa, Z. Boulkenafet, J. Komulainen, and A. Hadid, "Audiovisual synchrony assessment for replay attack detection in talking face biometrics," *Multimedia Tools and Applications*, vol. 75, no. 9, pp. 5329–5343, May 2016.

[6] J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in *Computer Vision, ACCV 2016 Workshops*. Cham: Springer International Publishing, 2017, pp. 251–263.

[7] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson, "3d convolutional neural networks for cross audio-visual matching recognition," *IEEE Access*, vol. 5, pp. 22 081–22 091, 2017.

[8] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 95:1–95:13, Jul. 2017. [Online]. Available: http://doi.acm.org/10.1145/3072959.3073640

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[10] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[11] F. K. Soong and A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 6, pp. 871–879, Jun. 1988.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

ISBN 978-90-827970-1-5 © EURASIP 2018