Tampered Speaker Inconsistency Detection with Phonetically Aware Audio-visual Features

Pavel Korshunov [1] Michael Halstead [1] Diego Castan [2] Martin Graciarena [2] Mitchell McLaren [2] Brian Burns [2] Aaron Lawson [2] Sebastien Marcel [1]

Abstract

The recent increase in social media based propaganda, i.e., 'fake news', calls for automated methods to detect tampered content. In this paper, we focus on detecting tampering in a video with a person speaking to a camera. This form of manipulation is easy to perform, since one can just replace a part of the audio, dramatically changing the meaning of the video. We consider several detection approaches based on phonetic features and recurrent networks. We demonstrate that by replacing standard MFCC features with embeddings from a DNN trained for automatic speech recognition, combined with mouth landmarks (visual features), we can achieve a significant performance improvement on several challenging publicly available databases of speakers (VidTIMIT, AMI, and GRID), for which we generated sets of tampered data. The evaluations demonstrate a relative equal error rate reduction of 55% (to 4.5% from 10.0%) on the large GRID corpus based dataset and a satisfying generalization of the model on other datasets.

1. Introduction

With the ubiquitous nature of social media, including Facebook and Instagram, creation and fast dissemination of high quality video content is increasing. Coupling these mediums with the constantly improving data tampering techniques, for both audio and visual content, facilitates the rise and the rapid spread of "fake news". Such deliberate misinformation through digital means is impacting political landscapes of several countries (Allcott & Gentzkow, 2017). Recent proposal to redesign Facebook with the aim to "create a more trustworthy platform"[1] demonstrates the importance of the fight against deliberate misinformation and tampering. It is therefore evident that the development of effective tools, which can automatically detect tampered audio-visual content, is of a paramount importance.

In this paper, we focus on detecting audio-visual tampering in a video of a speaking person, i.e., the inconsistencies between the video and audio tracks. This problem is rooted in dubbing and lip-syncing detections, but in case of tampering, there is a malicious intent to misrepresent the meaning of the video and to deceive the viewer into thinking that it is the original video.

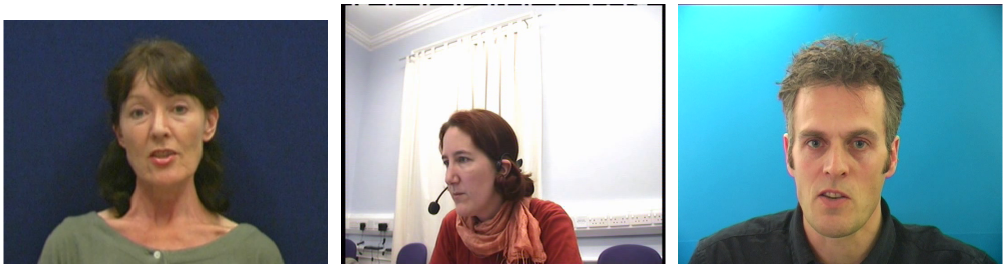
The majority of recent approaches (Sargin et al., 2007; Le & Odobez, 2016; Chung et al., 2016; Boutellaa et al., 2016; Chung & Zisserman, 2017; Torfi et al., 2017) for lip-syncing or dubbing detection focus on extracting separate feature sets for audio and video. For the audio component, mel-scale frequency cepstral coefficients (MFCC) are typically selected, however, visual features and classification methods vary considerably between techniques. The visual features range from hand selected techniques such as optical flow (Le & Odobez, 2016) to end-to-end feature extraction and classification processes, like the one used in (Chung et al., 2016). While a variety of classification techniques have been explored, best performing examples include long short-term memory (LSTM) (Chung et al., 2016) or convolutional neural networks (CNNs) (Torfi et al., 2017).

In this paper, we evaluate and expand on the previous work (Korshunov & Marcel, 2018). While using similar visual features, distances between detected mouth landmarks (inspired by (Suwajanakorn et al., 2017)), two different audio schemes are evaluated across varying classifiers and parameters on different databases with tampering attacks. To evaluate performance of the tampering detection systems, for the audio component, we use generic MFCC features or embeddings of a DNN trained for speech recognition. We also explore dimensionality reduction of blocks of features with principle component analysis (PCA) and varying LSTM sizes for classification.

[1] Idiap research Institute, Martigny, Switzerland [2] SRI, Menlo Park, CA, USA. Correspondence to: Pavel Korshunov <pavel.korshunov@idiap.ch>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] https://www.nytimes.com/2019/04/30/technology/facebook-private-communication-groups.html



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

Figure 1. Example screenshots from the three databases used in the experiments. The resolutions for each dataset are: VidTIMIT 512×384, AMI 720×576, and GRID 720×576.

We used three different publicly available databases with audio-visual data: VidTIMIT[2], AMI corpus[3], and GRID corpus[4], for which we have generated a tampered set of data where we randomly replaced the audio track of one subject with that of another. We focus only on audio channel tampering, because visually tampered data, e.g., by using 'Deepfakes', is scarce (Korshunov & Marcel, 2019) and its generation is out of the scope of the paper. To promote reproducibility, we provide the scripts for generation of tampered data and the implementations of evaluated systems as an open source Python package[5].

2. Databases and protocol

We have selected three different publicly available databases (see the example screenshots in Figure 1) with audio-visual data: VidTIMIT[6], AMI corpus[7], and GRID corpus[8]. For each video in a database, we generated between four and five tampered versions for the VidTIMIT and AMI datasets and a single tampered version for GRID. The tampered data was created by randomly replacing the audio track of one subject with that of another, while trying to preserve the same phrasing as much as possible. For instance, in VidTIMIT and GRID, the same or a similarly spoken phrase is the first replacement candidate.

It should be noted that VidTIMIT and GRID datasets were shot in a controlled environment (subjects facing the camera and reciting predetermined short phrases), while AMI dataset consists of informal meeting recordings (profile faces, mouth occlusions, unclear speech, etc.), which results in a more complex yet practical and realistic set of data. Also, the databases vary significantly in the amount of data available. VidTIMIT is an example of a 'toy' database of less than an hour of video in total, since there are only 10 of $3 - 6$ second videos recorded for each of the 43 subjects. GRID corpus is the largest database with the total length of 27 hours, and it has 1000 of 3 second videos for each of the 33 subjects. Finally, the AMI corpus is the most realistic representation of a practical scenario for tampering detection due to it containing recordings of office base meetings in an informal setting. Since in this paper, we assume a scenario of a single person talking to a camera, from the larger AMI corpus, we have selected 977 close up camera videos with a continues speech by one person, which resulted in approximately 6 hours of video in total for the 54 subjects with the average video length of 22 seconds.

The databases were split into training (*Train*) and development (*Test*) subsets and we arranged that the same subject would not appear in both sets. For more details about the databases, the tampering sets, and the way the data was split, please see our previous work (Korshunov & Marcel, 2018) and the scripts for generating tampered data [5].

We evaluate the tampering detection systems by computing scores for all videos in a dataset. For each possible threshold θ that separates genuine scores from tampered scores, we compute commonly used metrics for evaluation of classification systems: false acceptance rate (FAR) and false reject rate (FRR). Threshold at which these FAR and FRR are equal leads to an equal error rate (EER), which is commonly used as a single value metric of the system performance.

3. 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 a 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.

For the visual features, we use normalized distances between

[2] http://conradsanderson.id.au/vidtimit/
[3] http://groups.inf.ed.ac.uk/ami/download/
[4] http://spandh.dcs.shef.ac.uk/gridcorpus/
[5] https://gitlab.idiap.ch/bob/bob.paper.lipsync2019
[6] http://conradsanderson.id.au/vidtimit/
[7] http://groups.inf.ed.ac.uk/ami/download/
[8] http://spandh.dcs.shef.ac.uk/gridcorpus/

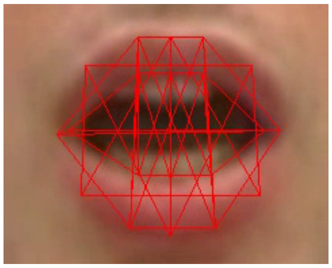


Figure 2. A screenshot of the mouth region from a GRID database video with detected visual features.

mouth landmarks (mouth deltas), extracted with OpenPose[9], which reliably performs a body pose (Cao et al., 2017) and face landmarks (similar to hands detection in (Simon et al., 2017)) detection. Using these landmarks, we first estimate whether the video shows a frontal face (e.g., AMI database contains a lot of profile faces). Then, to characterize the mouth movements, we compute 42 different distances between the 20 detected points of the mouth (see Figure 2 for an example). These features, while simple, have been shown to be quite effective, both in the previous work (Korshunov & Marcel, 2018) and for generating realistic mouth movements from a given audio track containing speech (Suwajanakorn et al., 2017). Due to the significant variation in mouth position in profile faces, in this work, we only consider frontal faces.

For the audio features, we consider two types of features: (i) mel-scale frequency cepstral coefficients (MFCC), which are a commonly used acoustic features used in various audio processing applications, and (ii) senone based DNN Embeddings – the language specific phonetic features that are successfully used in speaker verification (Lei et al., 2014). For the first type, we use 42 feature values, which consist of 13 MFCCs, their energy, delta, and double-deltas. They are computed from the 24 banks of the power spectrum (power of magnitude of 512-sized FFT) on 20ms-long windows with 10ms overlap. DNN Embeddings represent a 80-dimensional bottleneck layer of the network that targets 5302 senones (tri-phone states) learned using Fisher (Fis) and Switchboard data (Swi). The network has 5 hidden layers of 1200 nodes with the fourth hidden layer being the bottleneck of 80 nodes, and is constructed similar to (Lei et al., 2014). DNN Embeddings are also extracted for each 20ms-long window with 10ms overlap. In addition to types of audio features, we also used a *Combined* feature vector, in which we join MFCC based features and DNN Embeddings.

To preserve local temporal context, we combine features of the 0.2 seconds temporal block into one vector (Le & Odobez, 2016; Korshunov & Marcel, 2018). Since the video frame rate in our datasets is 25fps and the audio is sampled at 100fps, one such block has 5 visual and 20 audio features, which leads to $5 \times 42 + 20 \times 42 = 1050$ values, when MFCC features are used for audio. To reduce the dimensionality of the temporal block, we use principal component analysis (PCA). We used 60 (about 95% of variance depending on the database) components in the experiments with MFCC or DNN Embeddings and 100 components when used Combined features. PCA matrix is trained on the *Train* set of a database and then applied on the *Test* set.

3.1. System architecture

To learn the temporal sequence in the audio-visual streams, we use long short-term memory (LSTM) network implemented with Tensorflow[10] with a *tanh* activation function. We evaluate various cell sizes of the LSTMs and training batch sizes. The final classification stage contains a fully connected layer with two neurons corresponding to the two classes (genuine or tampered), this layer takes as input the last output of the LSTMs. A SoftMax cross entropy loss function is computed and Adam optimization algorithm (Kingma & Ba, 2014) 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 an LSTM. The size of the temporal window is also a parameter of the system (in this case statically set to 10 based on the previous work (Korshunov & Marcel, 2018)). The training data was balanced in terms of the number of samples from genuine and tampered classes. Since in VidTIMIT and AMI databases, the genuine set is much smaller than the tampered, the data from tampered set is randomly sampled to produce a similar number of features as in the genuine set. To improve the 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. Batch normalization is also performed on the input and the output of the LSTM. The LSTM size cell were varied as 32, 64, 128, and 256 and batch size 8 and 16 were used.

4. Evaluation results

The selected results on three databases are shown in Table 1, which presents EER for different systems computed on *Test* set for the systems trained on the same database. The 'System' column shows the LSTM cell size, a batch size of 8 was used as results indicated superior performance.

Table 1 demonstrates that DNN Embeddings outperform both the MFCCs and the Combined features. The greatest improvement is seen when training on the GRID database, where we achieve a relative improvement of 55% over the MFCC features (which we proposed in the previous work (Korshunov & Marcel, 2018)). While using the combi-

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

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

Table 1. EER values computed on *Test* set of the same database that was used for training. PCA 60 is used for MFCC and DNN Embeddings and PCA 100 for the Combined features.

Database	System	Audio	Test (%)
VidTIMIT	LSTM 64	MFCC	23.3
	LSTM 128	MFCC	21.3
	LSTM 32	**DNN Embeddings**	**17.4**
	LSTM 64	DNN Embeddings	18.8
	LSTM 128	DNN Embeddings	24.7
	LSTM 64	Combined	18.4
	LSTM 128	Combined	22.7
AMI	LSTM 64	MFCC	38.9
	LSTM 128	MFCC	34.7
	LSTM 64	DNN Embeddings	31.2
	LSTM 128	**DNN Embeddings**	**27.4**
	LSTM 64	Combined	30.3
	LSTM 128	Combined	30.4
GRID	LSTM 64	MFCC	10.4
	LSTM 128	MFCC	10.0
	LSTM 64	DNN Embeddings	4.7
	LSTM 128	**DNN Embeddings**	**4.5**
	LSTM 64	Combined	5.5
	LSTM 128	Combined	5.0

nation of features does not improve performance, they still outperform MFCCs.

Table 1 also shows that the problem of audio-visual tampering detection is far from being solved. The lowest EER is achieved on GRID dataset, which has many videos shot in a controlled environment with subjects facing the camera (no profile faces) speaking clearly predetermined short phrases. GRID is the largest and the most 'academic' database and that is probably why we can reach an EER of 4.5%.

The most challenging and also the most realistic dataset is AMI. Many AMI videos have profile faces, on which our profile face detector fails. Also, AMI subjects have a free conversation in uncontrolled office environment, the dataset is poorly annotated, speakers are often interrupted, and some portions of the video can show one person but have an audio from another person behind the camera. Such complex genuine data poses additional challenges for tampering detection and leads to a poor performance.

Relative to GRID, the poorer performance on VidTIMIT is probably due to the small size of VidTIMIT. However, even this small database demonstrates the importance of selecting appropriate features, since, with all the other parameters being the same, using DNN Embeddings leads to a considerably smaller EERs compared to using MFCCs.

To demonstrate the importance of training tampering detection systems on the large dataset and to understand how such systems generalize across different data, we have evaluated systems trained on GRID dataset by testing them on other two datasets. The cross-database evaluation results are shown in Table 2. It should be noted that the systems evaluated in this table are the same systems as GRID-based systems shown in Table 1. The difference in Table 2 is that the *Test* set is from VidTIMIT or AMI datasets.

Table 2. Systems are trained on GRID (*Train* set) and evaluated on *Test* sets of VidTIMIT and AMI. PCA 60 is used for MFCC and DNN Embeddings and PCA 100 for the Combined features.

Train DB	Test DB	System	Audio	Test (%)
Grid	VidTIMIT	LSTM 64	MFCC	26.6
Grid	VidTIMIT	LSTM 128	MFCC	27.2
Grid	VidTIMIT	LSTM 64	DNN Embeddings	15.1
Grid	**VidTIMIT**	**LSTM 128**	**DNN Embeddings**	**9.8**
Grid	VidTIMIT	LSTM 64	Combined	13
Grid	VidTIMIT	LSTM 128	Combined	14
Grid	AMI	LSTM 64	MFCC	39.0
Grid	AMI	LSTM 128	MFCC	39.0
Grid	AMI	LSTM 64	DNN Embeddings	35.9
Grid	AMI	LSTM 128	DNN Embeddings	36.8
Grid	**AMI**	**LSTM 64**	**Combined**	**31.3**
Grid	AMI	LSTM 128	Combined	35.9

Table 2 demonstrates that the system trained on large enough GRID dataset can lead to a considerably lower EER for a 'small' VidTIMIT dataset. The EER of the best performing system in Table 2 for VidTIMIT is 9.8%, which is almost twice lower than 17.4% when trained on VidTIMIT itself (see Table 1). The situation is not as impressive with cross-evaluation on AMI dataset with the best EER of 31.3% in Table 2 being higher than 27.4% of the system trained on AMI. Again, this can be explained by the fact that the AMI data is very different. For a GRID-trained system, the AMI data can be considered as a type of 'unseen' data, which detection is challenging in many domains, including face anti-spoofing (Arashloo et al., 2017).

5. Conclusions

In this paper, we propose the system for detecting audio-visual inconsistencies in videos of speaking people based on the audio-visual features that characterize physical properties of human speech. We compare these features with the commonly used features such as MFCCs and show that more tailored featured lead to better performance. The evaluations were done on three publicly available databases, namely, VidTIMIT, AMI, and GRID, which we augmented with tampered video data.

The results also demonstrate that with a large enough training dataset that more realistically reflects the practical video examples, the system can generalize even on an unseen data. We still lack a more challenging and realistic public database with tampered data.

Therefore, in the future, we will focus on large-scale databases with more sophisticated tampering data, including replacing single words and face swapping with GANs, exploring more complex architectures and their generalization properties, and features invariant to facial rotation.

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 are not representing the official views or policies of AFRL or DARPA.

References

Fisher english training speech part 1 transcripts. https://catalog.ldc.upenn.edu/LDC2004T19. Accessed: 2019-04-30.

Switchboard-1 telephone speech corpus. https://www.isip.piconepress.com/projects/switchboard/. Accessed: 2019-04-30.

Allcott, H. and Gentzkow, M. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017. doi: 10.1257/jep.31.2.211.

Arashloo, S. R., Kittler, J., and Christmas, W. An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE Access*, 5: 13868–13882, 2017.

Boutellaa, E., Boulkenafet, Z., Komulainen, J., and Hadid, A. Audiovisual synchrony assessment for replay attack detection in talking face biometrics. *Multimedia Tools and Applications*, 75(9):5329–5343, May 2016.

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

Chung, J. S. and Zisserman, A. Out of time: Automated lip sync in the wild. In *Computer Vision, ACCV 2016 Workshops*, pp. 251–263, Cham, 2017. Springer International Publishing. ISBN 978-3-319-54427-4.

Chung, J. S., Senior, A. W., Vinyals, O., and Zisserman, A. Lip reading sentences in the wild. *CoRR*, abs/1611.05358, 2016. URL http://arxiv.org/abs/1611.05358.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Korshunov, P. and Marcel, S. Speaker inconsistency detection in tampered video. In *European Signal Processing Conference (EUSIPCO)*, pp. 2375–2379, September 2018.

Korshunov, P. and Marcel, S. Deepfakes: a new threat to face recognition? assessment and detection. *CoRR, arXiv*, abs/1812.08685, 2019. URL http://arxiv.org/abs/1812.08685.

Le, N. and Odobez, J.-M. Learning multimodal temporal representation for dubbing detection in broadcast media. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pp. 202–206, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3603-1. doi: 10.1145/2964284.2967211. URL http://doi.acm.org/10.1145/2964284.2967211.

Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 1695–1699. IEEE, 2014.

Sargin, M. E., Yemez, Y., Erzin, E., and Tekalp, A. M. Audiovisual synchronization and fusion using canonical correlation analysis. *IEEE Transactions on Multimedia*, 9(7):1396–1403, Nov 2007. ISSN 1520-9210. doi: 10.1109/TMM.2007.906583.

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

Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 36(4):95:1–95:13, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073640. URL http://doi.acm.org/10.1145/3072959.3073640.

Torfi, A., Iranmanesh, S. M., Nasrabadi, N., and Dawson, J. 3d convolutional neural networks for cross audio-visual matching recognition. *IEEE Access*, 5:22081–22091, 2017. doi: 10.1109/ACCESS.2017.2761539.