Toward End-to-End Deception Detection in Videos

Hamid Karimi
Computer Science and Engineering
Michigan State University
East Lansing, USA
karimiha@msu.edu

Jiliang Tang
Computer Science and Engineering
Michigan State University
East Lansing, USA
tangjili@msu.edu

Yanen Li
Sanp. Inc
Los Angeles, USA
yanen.li@snap.com

***Abstract*—There are various real-world applications such as video ads, airport screenings, courtroom trials, and job interviews where deception detection can play a crucial role. Hence, there are immense demands on deception detection in videos. However, videos are inherently complex; moreover, they lack detective labels in many real-world applications, which poses tremendous challenges to traditional deception detection methods. In this paper, we study the problem of deception detection in videos. In particular, we provide a principled way to capture rich information into a coherent model and propose an end-to-end framework DEV to detect DEceptive Videos automatically, which is robust to the small number of training data. Experimental results on real-world videos demonstrate the effectiveness of the proposed framework and further experiments are conducted to understand important factors of deception detection in videos.**

***Index Terms*—Deception Detection, Video, Audio, Limited Data, Interpretability**

I. Introduction

In various applications, deception detection in videos is very important. For instance, transportation security agents sometimes interview passengers for security purposes where identifying deception is paramount; detecting deception in police interrogation videos and recorded courtroom trials are very crucial for settling a case, which sometimes is a matter of someone's life and death; job interviews [1], transportation video surveillance [2], and many other applications can also benefit from deception detection in videos.

On the one hand, deception detection in videos faces three tremendous challenges. First, video data is inherently complex and naturally multi-modal while containing complicated temporal correlations. The complexity does not stop here. Innate complexity of deception itself makes deception detection in videos even more complex. Second, to understand deception in videos in more depth, identifying cues related to deception is necessary. Therefore, a deception detection model is desired to offer an interpretable solution. However, given the complexity of videos in general and deception itself in particular, having an efficient interpretable model for deception detection in videos is another challenge. Third, most videos in real-world applications lack the deceptive labels. As the efficiency of laymen for discerning detection is very low [3], labeling deceptive videos requires domain experts and is quite costly and effort consuming. Therefore, labeled videos are extremely limited, which further exacerbates the modeling challenge. On the other hand, videos offer great opportunities for understating deception. They contain two modes of data (i.e., vocal and visual modes), each of which can contribute to revealing deception cues. Moreover, the temporal cues hidden in a sequence of frames of a video could be quite informative for deception detection.

In this paper, we embrace challenges and opportunities to study deception detection in videos. In particular, we propose an end-to-end framework DEV, which can accurately detect **DE**ceptive **V**ideos automatically. DEV provides a unified solution to address the aforementioned challenges simultaneously – (1) it automatically captures rich and sophisticated features from complicated video data; (2) it delivers interpretable visual cues in deceptive and truthful videos; and (3) it is robust against the small number of training data. Experimental results on real-world videos demonstrate the effectiveness of the proposed framework DEV in detecting deceptive videos especially when the number of labeled videos is small. Comprehensive experiments are also conducted to understand the working of the proposed framework.

The rest of the paper is organized as follows. In Section II, we describe the solutions to address the challenges facing deception detection in videos and introduce the details of the proposed framework. Section III describes the experiment, results, and discussions. In Section IV, we briefly review the related work. We conclude the paper in Section V.

II. The Proposed Framework

Deception detection in videos faces several challenges – (a) video data, as well as the deception itself, are inherently complicated and also there are complex temporal dependencies in a video; (b) interpretable cues of deception should be delivered; and (c) typically limited labeled data is available. To solve these challenges, we propose an end-to-end framework DEV for deception detection in videos, which is demonstrated in Figure 1. It consists of three major model components – (1) an automate feature extraction component to address the challenge (a); (2) a visual interpretability component based on an attention mechanism to address (b); and (3) a classification component based on a metric learning approach for tackling (c). Next, we will detail each component.

A. Automated Feature Extraction

Video data is inherently complicated. Naturally, there are two modes of data in videos, namely vocal and visual modes.

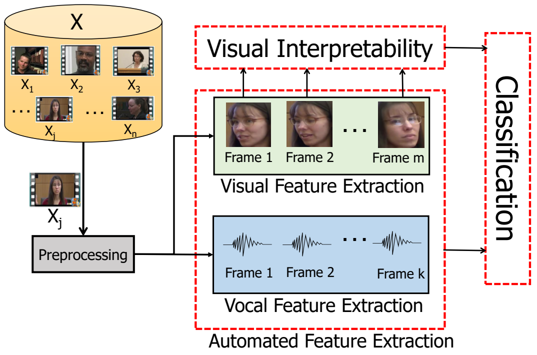


Fig. 1: The proposed end-to-end framework for deception detection in videos

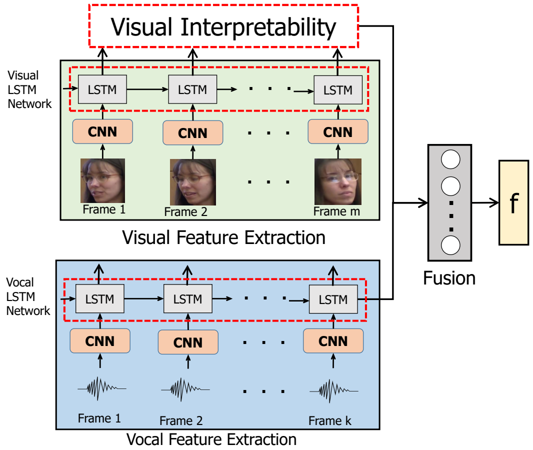


Fig. 2: Automated features extraction for deception detection in videos

In both modes, there exist local patterns in distinct portions of videos as well as temporal correlations among the portions. These sophisticated features (patterns) need to be captured effectively which is less likely to be feasible if hand-crafted features are employed. Hence, the complexities of both video data and deception make feature extraction for deception detection in videos quite challenging.

To address this challenge, we propose an automated feature extraction component demonstrated in Figure 2. As a video file is usually a large data sample, it is broken into a set of frames. This is performed on both visual and vocal modes as shown in Figure 2. First, we need to extract features from individual frames. To this end, we utilize the Convolutional Neural Network (CNN). CNNs have shown excellent results in computer vision applications [4], [5] and are increasingly used for audio-related applications [6]. What mostly makes a CNN as a successful model is its capability to extract features via applying various filters (weight matrices) to a region of data. We hypothesize that the filters in a deep CNN can capture visual deception cues. The same reasoning applies to vocal mode as well. Hence, a well-designed and well-trained CNN should be able to detect vocal an visual features relevant to deception

Now, suppose $j \in [1, n]$ is the index of video j, k is the length of an audio frame sequence (zero-padded to have length k if necessary), m is the length of an image frame sequence (zero-padded to have length m if necessary), a_i^j $(1 \leq i \leq k)$ is the output of the vocal CNN for audio frame i, and v_i^j is the output of the visual CNN for image frame i $(1 \leq i \leq m)$. Then, we use $A_j = \{a_1^j, a_2^j, \cdots, a_k^j\}$ and $V_j = \{v_1^j, v_2^j, \cdots, v_m^j\}$ to denote sequences of CNN outputs for vocal and visual modes, respectively.

Next, we need to investigate an entire sequence of features extracted from image/audio frames by visual/vocal CNNs. This requirement comes from the fact that usually in deceptive/truthful videos a single frame alone cannot determine the relevant cues. Rather, we might need to temporally inspect an entire video or part of it to identify deception cue(s). For instance, increased blinking shown as a possible cue of deception [7] can only be identified from a sequence of images containing the blinking process. This is similarly valid for the vocal cues. Hence, to capture temporal correlations in a sequence of features, we utilize Long-Short Term Memory (LSTM) [8] network which has shown success in many video applications [9]. An LSTM network effectively propagates information throughout a sequence of inputs and can identify temporal correlations in sequential data types like videos. This capability makes LSTM network a promising candidate for capturing temporal correlations in deceptive videos.

The input to the vocal/visual LSTM network is a sequence of features from their corresponding CNNs i.e., A_j and V_j. Let $Q_j = \{q_1^j, q_2^j, \cdots, q_k^j\}$ and $R_j = \{h_1^j, h_2^j, \cdots, h_m^j\}$ denote outputs of vocal and visual LSTM networks, respectively[1]. Output vectors R_j are passed to visual interpretability component to focus the model's attention on the relevant frame(s) aiming at delivering an interpretable solution. More details are presented in the next subsection.

B. Visual Interpretability

To better understand deception, a deception detection model needs to be interpretable. In this work, we focus on the interpretability of visual mode and leave the vocal interpretability for future work. One way to deliver an interpretable solution is through identifying an image frame that is most responsible for deception/truthfulness. In other words, for an interpretable model we need to identify where a subject exhibits more signs of deception by investigating features extracted from the visual LSTM network. To achieve this, we utilize an attention mechanism. Attention mechanisms have shown significant improvement in many deep neural network models [11]. In the proposed attention mechanism for the visual interpretability component, the final feature vector of the visual mode is a linear combination of all outputs of visual LSTM network as shown in the following:

[1] Due to the space constraints, we have not included CNN and LSTM formulations. Interested readers are referred to [10] and [8].

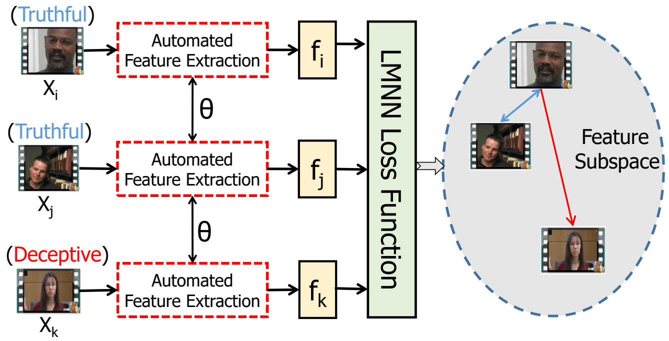


Fig. 3: Classification component of deception detection in videos. Three subnetworks share the parameters (θ).

$$
\begin{aligned}
g_j &= \sum_{i=1}^{m} s_i h_i^j, \quad \forall j \in [1,n] \\
s_i^j &= \frac{e^{u_i^j}}{\sum_{l=1}^{m} e^{u_l^j}} \quad \forall j \in [1,n], i \in [1,m] \\
u_i^j &= \mathcal{G}(w_i^T \times h_i^j + b_i) \quad \forall j \in [1,n], \forall i \in [1,m]
\end{aligned} \tag{1}
$$

where g_i is the final feature vector of visual feature extraction module in Figure 2, s_i is the attention score of the image frame i. The attention scores specify the contribution of each image frame at a video's final visual feature vector g_j and they should form a probability distribution. Therefore, each s_i^j is normalized by the softmax function. In Eq. 1, u_i^j is the unnormalized attention score of image frame i, $\mathcal{G}$ is a non-linear activation function, $||$ denotes the concatenation operation, W is a weight matrix, and b is a bias vector.

To deliver an interpretable output for a particular video, we retrieve the image frame corresponding to the maximum attention score. Afterward, through visualizing CNN outputs of that image, we demonstrate possible cues of deception on the image.

Ultimately, the final feature vector f_j used for classification is computed as follows.

$$
f_j = \mathcal{G}(W[g_j || q_k^j] + b) \tag{2}
$$

Eq. 2 represents a simple fusion mechanism combining the final vocal and visual features and it is demonstrated as *Fusion* in Figure 2.

C. Classification

To obtain a good classifier for deception detection, we typically need sufficient training samples. However, as mentioned before, the number of labeled deception/truthful videos is limited. To address the problem of limited training data, we change our perspective – we look into similarity and dissimilarity between samples instead of directly learning a classifier from training samples. Under the new perspective, the basic task is to learn a feature subspace wherein samples from the same class are close to each other and samples from different classes are further away from each other. Weinberger and Saul [12] followed a similar idea and proposed a method known as Large Margin Nearest Neighbor (LMNN). LMNN learns a metric that is of interest of k-Nearest Neighbor (kNN) rule: assign a new sample to the class of majority among k nearest neighbors. In this work, we adopt a similar idea and describe it in the following.

Figure 3 illustrates the classification component of the proposed approach. It takes a triplet of feature vectors from the automated feature extraction component. Feature vectors are retrieved from three identical sub-networks illustrated in Figure 2. The three sub-networks share the same parameters because we seek to find a common subspace which is unlikely to happen if three distinct networks are used. In addition, sharing the parameters can significantly reduce the number of parameters of the model. The classification component uses the following loss function to learn a proper distance metric suitable for kNN.

$$
\begin{aligned}
\mathcal{L} = &\sum_{\forall i,j} ||f_i - f_j||_2^2 + \\
&\beta \sum_{\forall i,j,k} max(0, [1 + ||f_i - f_j||_2^2 - ||f_i - f_k||_2^2])
\end{aligned} \tag{3}
$$

where i and j are indexes of a pair of videos from the same class, k is the index of a video from a different class, β is a positive number, and $||.||_2$ is Euclidean distance. Here f is the non-linear transformation of a video's features. The first term in Eq. 3 is the *pull* term which pulls similarly labeled samples (videos) together and the second term is the *push* term which pushes dissimilar samples apart. The latter enforces a margin of unit size between dissimilar samples as well. *Since now we accept triplets as inputs instead of individual samples, the number of the potential training samples under the new perspective significantly increases, which can mitigate the limited number of training data in the original problem.* In fact, considering all possible triplets as well as minimizing the loss function of Eq. 3 help in exploring the underlying structure of the data samples in a better way without the need of a huge number of training samples.

III. Experiments

In this section, we first introduce the dataset followed by the experimental settings and finally present the results with their discussions.

A. Dataset

We use the real trial videos introduced in [13] for evaluating the framework. The video clips have been human annotated as either deceptive or truthful based on the court verdicts (i.e., guilty, non-guilty, and exoneration), the verification of the police report against a suspect's presented statements, and so on. The original dataset includes 121 short video clips including 60 truthful and 61 deceptive clips. Audio and image frames are extracted from each video with the frame rate of 1 frame/second. The maximum length of audio and image frame sequences (k and m in Figure 2) are 81 and 79, respectively.

B. Experimental Settings

We use 10 random videos from each class for the test set and the remaining videos are used for the training set. Tensorflow 1.3.0 [2] is used for the implementation. We use cross-validation to tune the hyperparameters where L2 regularization coefficient is set to 0.01, dropout to 0.5, the number of hidden units of visual/vocal LSTM networks to 200, and parameter β in Eq. 3 to 0.4. The learning rate is getting decreased dynamically starting from 0.01 with decaying factor of 0.9 after each epoch. The experiments are conducted for 5 epochs. In each epoch, all possible triplets are selected for training in the batch size of 32 triplets. When the training process finishes, we save the model and use it for predicting the test samples and reporting the performance of the framework. We opt for accuracy as the performance metric in our evaluation because the dataset is very balanced and we found f-measure performs similarly as accuracy.

C. Results and Discussions

In order to evaluate the performance of the proposed framework DEV, we conduct a set of experiments. In line with the challenges mentioned before, we seek to answer the following research questions.

- (Q1). How does the proposed framework perform on deception detection in videos?
- (Q2). Is the framework robust against the small number of training videos?
- (Q3). How does the framework provide an interpretable solution?
- (Q4). What are the contributions of the framework's components in video deception detection?

1) Comparison: To answer the first two questions, (i.e. Q1 and Q2), we compare the proposed framework DEV with the following representative baseline methods:

- **ISO9**. In this baseline method, we use Interspeech 2009 (IS09) emotion challenge feature set[3]. It consists of 384 vocal features which have been shown to be predictive of deception [14]. This baseline only uses vocal information.
- **IS13**. This is similar to ISO9 but utilizes Interspeech 2013 (IS13) ComParE Challenge features instead[4] and includes 6373 vocal features. This baseline also only utilizes vocal information.
- **AUs**. For this method, we use 18 facial Action Units (AUs) [15], which have been shown to be informative for deception detection [16]. We extract AUs using OpenFace [17]. This baseline only uses visual information
- **AUs+IS09**. As a hybrid approach, we combine vocal IS09 and visual AUs features.
- **AUs+IS13**. As another hybrid approach, we combine vocal IS13 and visual AUs features.

TABLE I: Overall classification accuracy comparison

Source	Method	Accuracy (%)
–	Random	50
Vocal	IS09	71.5
	IS13	70.05
	DEV-vocal	**74.16**
Visual	AUs	68.5
	DEV-visual	**75.00**
Hybrid	Vocal: IS09 Visual: AUs	73.5
	Vocal: IS13 Visual: AUs	72.1
	DEV	**84.16**

- **DEV-vocal**. It is a variant of the proposed framework, which only uses vocal information while setting the parameters of the visual part to zero.
- **DEV-visual**. It is a variant of the proposed framework, which only uses visual information while setting the parameters of the vocal part to zero.

For the baseline methods, we use RandomForest classifier and tune the hyperparameters using the cross-validation. Baseline methods and our model (i.e. DEV) are repeated 10 times, and the average accuracy on the test set is reported for each method. Table I shows the detection accuracy results. First of all, as the number of deceptive and truthful videos in the test set are equal, the random guess is 50%. Based on the results presented in Table I, we make the following observations:

- In general, hybrid methods achieve better performance than single modes. This observation supports that both vocal and visual information contain complementary information for deception detection.
- The variants of the proposed framework i.e., DEV-vocal and DEV-visual, outperform the corresponding baselines with only vocal information or visual information. These results suggest that automated feature extraction component learns powerful features for deception detection. It also shows the robustness of the model against the absence of one mode of data.
- The proposed framework can accurately detect deceptive videos by incorporating vocal and visual information coherently even with a small number of training data.

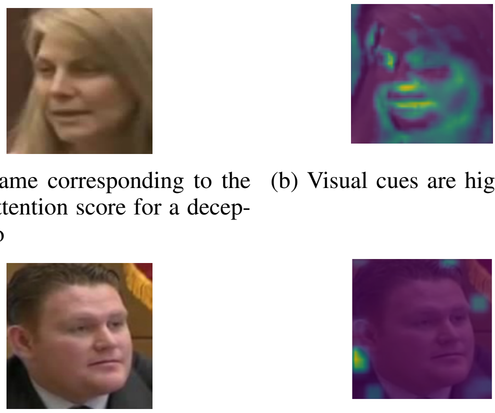
2) Model Interpretability (a case study): Understanding deception cues is of great importance. In Figure 4, we show a case study to illustrate the interpretability power of the proposed framework. In this case study, we apply the visual interpretability component on two test videos, one from each class. To do this, we first retrieve the probabilities of associated with visual frames for both videos i.e., s_i^j attention scores in Eq. 1. Then a frame corresponding to the maximum probability is selected out. Afterward, we visualize the activations of the last convolution layer in the visual CNN. Based on the results of the case study presented in Figure 4, we make the following observations:

- There is a striking difference between the cues activated in the deceptive video shown in Figure 4b and those of the

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

[3] http://emotion-research.net/sigs/speech-sig/emotion-challenge

[4] http://emotion-research.net/sigs/speech-sig/is13-compare



(a) An frame corresponding to the highest attention score for a deceptive video

(b) Visual cues are highlighted

(c) An frame corresponding to the highest attention score for a truth video

(d) Visual cues are highlighted

Fig. 4: A case study of the model visual interpretability

truthful video shown in Figure 4d. In the truthful case, the subject's face has not shown any noticeable cues while in the deceptive video, certain regions of the subject's face have been highlighted. This corroborates the fact that the proposed model effectively and automatically detects indicative cues of deception.

- Facial regions activated in the deceptive case shown in Figure 4b suggest possible regions related to deception. Several of these facial regions such as lips and eyebrows have been previously shown to be related to deception [13], [18]. Interestingly though, a few other regions such as those around the subject's nostrils or cheeks have been activated as well. This signifies the fact that the proposed model optimally and comprehensively inspects the various regions of an image aiming at finding the visual cues related to deception. We leave more investigation to future work.

3) Impact of the Model Components: As described before, the proposed framework contains three important components: the automated feature extraction, the visual interpretability, and the classification. The research question `(Q4)` is concerned with the impact of these components on deception detection in videos. To answer this question, three experiments are conducted as described in the following.

First, we evaluate the effectiveness of LSTM networks. To achieve this, we keep LSTM networks but shuffle their input sequences randomly (both audio and video frames). Figure 5a shows the result. It can be observed that shuffling the frames randomly reduces the performance, which supports the importance of temporal correlations for deception detection in videos.

In the second experiment, we evaluate the performance of the visual interpretability component. To do so, the visual interpretability component is not used. More specifically, all visual frames have identical attention score (refer to Eq. 1). The result of this experiment is shown in Figure 5b. The performance with visual interpretability is better than when the visual interpretability is omitted.

In the third experiment, we evaluate the performance of the classification component. We change our perspective to train a direct discriminative classifier in the training process instead of learning a metric. The cross-entropy loss function replaces the LMNN loss function of Eq. 3. The result is shown in Figure 5c. As shown in the figure, classification component using triplet sampling strategy and LMNN loss function significantly outperform its cross-entropy counterpart.

IV. Related Work

Generally, there are two broad categories of deception cues: verbal and non-verbal [7]. Verbal cues are concerned with lexical features such as unigrams, Linguistic Inquiry Word Count (LIWC) [19] lexicons, Part-Of-Speech (POS) tags, etc. Non-verbal cues include physiological cues (e.g., facial blood flow), vocal cues (e.g., pause and silence in voice), and visual cues (e.g., scowling) [20].

Liars usually tend to use a language different from that of truth-tellers. This is the intuition for using linguistic features for deception detection [7], [21]–[23]. Despite the success of linguistic approaches, they are language-dependent which cannot be easily transferred to another language.

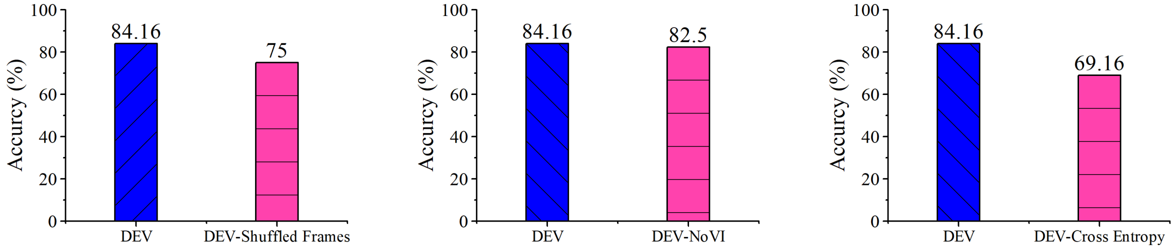
Another direction for deception detection is studying speech signals. Some of the cues in this category include sound pitch, pause, intensity, and so on [24]. Benus et al. [25] found the correlation between pauses and truthful speeches. Levitan et al. [26] constructed a dataset of English and Mandarin speeches. They discovered that along with vocal features, the individual difference (gender, native languages, etc) are also indicative in determining deceptive speeches.

In addition to the vocal features, visual cues such as facial expressions [27], body part movement [28], and head movement [28] have been shown to be informative for automatic deception detection in videos. Compared to the linguistic and vocal studies, however, there exists a limited number of works in this area. The main reason is that labeling deceptive/truthful videos is a time and effort consuming process.

Multimodal studies have recently gained interest in automatic deception detection. In [29], the authors combined speech features and lexical features to automatically detect deception in audios. Mendels et al. [14] combined vocal and lexical features in a deep model for an automatic deception detection. Similar to our work, authors of [16] extracted visual and vocal features for deception detection. However, unlike us, they found out that the vocal features are not very effective compared to other sources.

V. Conclusion

In this paper, we proposed an end-to-end framework DEV for video deception detection. This framework helped solve several challenges – (1) complexity of deception and video; (2) the need to offer an interpretable solution; and (3) the small number of training data. The experimental results demonstrated that (1) the proposed framework outperforms



(a) Automated feature extraction (b) Visual interpretability (c) Classification

Fig. 5: Evaluation results of component analysis experiments

representative baselines; (2) acoustic and visual information are complementary; and (3) the proposed framework is robust to the small number of training data. As future work, first, we plan to construct a larger dataset for deception detection, preferably from real-life videos. We also plan to investigate vocal interpretability i.e., discovering relevant acoustic deception cues in a data-driven manner.

Acknowledgment

This work is supported by the National Science Foundation (NSF) under grant numbers IIS-1714741, IIS-1715940 and CNS-1815636.

References

[1] B. Weiss and R. S. Feldman, "Looking good and lying to do it: Deception as an impression management strategy in job interviews," *Journal of Applied Social Psychology*, vol. 36, no. 4, pp. 1070–1086, 2006.

[2] M. L. Jensen, T. O. Meservy, J. Kruse, J. K. Burgoon, and J. F. Nunamaker, "Identification of deceptive behavioral cues extracted from video," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE, 2005, pp. 1135–1140.

[3] C. F. Bond Jr and B. M. DePaulo, "Accuracy of deception judgments," *Personality and social psychology Review*, vol. 10, no. 3, pp. 214–234, 2006.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[7] B. M. DePaulo, J. J. Lindsay, B. E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper, "Cues to deception." *Psychological bulletin*, vol. 129, no. 1, p. 74, 2003.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*, 2015, pp. 843–852.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[13] V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 59–66.

[14] G. Mendels, S. I. Levitan, K.-Z. Lee, and J. Hirschberg, "Hybrid acoustic-lexical deep learning approach for deception detection."

[15] P. Ekman and W. V. Friesen, "Facial action coding system," 1977.

[16] M. Jaiswal, S. Tabibu, and R. Bajpai, "The truth and nothing but the truth: Multimodal analysis for deception detection," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 938–943.

[17] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–10.

[18] Z. Wu, B. Singh, L. S. Davis, and V. Subrahmanian, "Deception detection in videos," *arXiv preprint arXiv:1712.04415*, 2017.

[19] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[20] E. Fitzpatrick, J. Bachenko, and T. Fornaciari, "Automatic detection of verbal deception," *Synthesis Lectures on Human Language Technologies*, vol. 8, no. 3, pp. 1–119, 2015.

[21] L. Zhou, J. K. Burgoon, J. F. Nunamaker, and D. Twitchell, "Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications," *Group decision and negotiation*, vol. 13, no. 1, pp. 81–106, 2004.

[22] H. Karimi, P. Roy, S. Saba-Sadiya, and J. Tang, "Multi-source multi-class fake news detection," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1546–1557.

[23] H. Karimi, C. VanDam, L. Ye, and J. Tang, "End-to-end compromised account detection," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 314–321.

[24] J. Hirschberg, S. Benus, J. M. Brenier, F. Enos, S. Friedman, S. Gilman, C. Girand, M. Graciarena, A. Kathol, L. Michaelis *et al.*, "Distinguishing deceptive from non-deceptive speech," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[25] S. Benus, F. Enos, J. Hirschberg, and E. Shriberg, "Pauses in deceptive speech."

[26] S. I. Levitan, M. Levine, J. Hirschberg, N. Cestero, G. An, and A. Rosenberg, "Individual differences in deception and deception detection."

[27] Y.-L. Tian, T. Kanade, and J. F. Cohn, "Facial expression analysis," *Handbook of face recognition*, pp. 247–275, 2005.

[28] R. Mihalcea and C. Strapparava, "The lie detector: Explorations in the automatic recognition of deceptive language," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 2009, pp. 309–312.

[29] S. I. Levitan, G. An, M. Ma, R. Levitan, A. Rosenberg, and J. Hirschberg, "Combining acoustic-prosodic, lexical, and phonotactic features for automatic deception detection." 2016.